# REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

| | | |
|---|---|---|
| 1. AGENCY USE ONLY *(Leave blank)* | 2. REPORT DATE<br>August 1991 | 3. REPORT TYPE AND DATES COVERED<br>memorandum |

4. TITLE AND SUBTITLE

Affine Mathcing With Bounded Sensor Error: A Study of Geometric Hashing and Alignment

5. FUNDING NUMBERS

N00014-86-K-0685
DACA76-85-C-0010
N00014-85-K-0124
IRI-8900267
IRI-9057928

6. AUTHOR(S)

W. Eric L. Grimson, Caneil P. Huttenlocher and David W. Jacobs

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Artificial Intelligence Laboratory
545 Technology Square
Cambridge, Massachusetts 02139

8. PERFORMING ORGANIZATION REPORT NUMBER

AIM 1250

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
Information Systems
Arlington, Virginia 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

None

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution of this document is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

**Abstract.** Affine transformations of the plane have been used in a number of model-based recognition systems, in order to approximate the effects of perspective projection. The mathematics underlying these methods is for *exact* data, where there is no positional uncertainty in the measurement of feature points. In practice, various heuristics are used to adapt the methods to real data with uncertainty. In this paper, we provide a precise analysis of affine point matching under uncertainty. We obtain an expression for the range of affine-invariant values that are consistent with a given set of four points, where each data point lies in a disk of radius $\epsilon$. This analysis reveals that the range of affine-invariant values depends on the actual $x$-$y$-positions of the data points. That is, when there is uncertainty in the data then

(continued on back)

14. SUBJECT TERMS (key words)

object recognition
affine matching

15. NUMBER OF PAGES

52

16. PRICE CODE

| | | | |
|---|---|---|---|
| 17. SECURITY CLASSIFICATION OF REPORT<br>UNCLASSIFIED | 18. SECURITY CLASSIFICATION OF THIS PAGE<br>UNCLASSIFIED | 19. SECURITY CLASSIFICATION OF ABSTRACT<br>UNCLASSIFIED | 20. LIMITATION OF ABSTRACT<br>UNCLASSIFIED |

Block 13 continued:

the representation is no longer invariant with respect to the Cartesian coordinate system. This is problematic for the geometric hashing method, because it means that the precomputed lookup table used by that method is not correct when there is positional uncertainty in the sensor data. We analyze the effect that this has on the probability that the geometric hashing method will find false positive matches of a model to an image, and contrast this with a similar analysis of the alignment method.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1250 August, 1991

# Affine Matching With Bounded Sensor Error: A Study of Geometric Hashing and Alignment

W. Eric L. Grimson, Daniel P. Huttenlocher[1] & David W. Jacobs

**Abstract.** Affine transformations of the plane have been used in a number of model-based recognition systems, in order to approximate the effects of perspective projection. The mathematics underlying these methods is for *exact* data, where there is no positional uncertainty in the measurement of feature points. In practice, various heuristics are used to adapt the methods to real data with uncertainty. In this paper, we provide a precise analysis of affine point matching under uncertainty. We obtain an expression for the range of affine-invariant values that are consistent with a given set of four points, where each data point lies in a disk of radius $\epsilon$. This analysis reveals that the range of affine-invariant values depends on the actual $x$-$y$-positions of the data points. That is, when there is uncertainty in the data then the representation is no longer invariant with respect to the Cartesian coordinate system. This is problematic for the geometric hashing method, because it means that the precomputed lookup table used by that method is not correct when there is positional uncertainty in the sensor data. We analyze the effect that this has on the probability that the geometric hashing method will find false positive matches of a model to an image, and contrast this with a similar analysis of the alignment method.

Acknowledgments: This report describes research done in part at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by an Office of Naval Research University Research Initiative grant under contract N00014-86-K-0685, and in part by the Advanced Research Projects Agency of the Department of Defense under Army contract number DACA76-85-C-0010 and under Office of Naval Research contract N00014-85-K-0124. WELG is supported in part by NSF contract number IRI-8900267. DPH is supported at Cornell University in part by NSF grant IRI-9057928 and matching funds from General Electric and Kodak.

[1] Dept. of Comp. Sci., Cornell University, Ithaca, NY

# 1 Introduction

Object recognition is a central problem in computer vision, and model-based methods constitute one prevalent approach to this problem. In the model-based approach, a set of geometric features that constitute a model of an object are compared against like features that have been extracted from an image of a scene (cf. [3, 7]). The process of comparing a model with an image generally involves determining a valid correspondence between a subset of the model features and a subset of the features found in the image. In order for such a correspondence to be valid, it is usually required that there exist some transformation of a given type mapping each model feature onto its corresponding image feature. This transformation generally specifies the *pose* of the object – its position and orientation with respect to the image coordinate system. The quality of a given hypothesized transformation is then evaluated based on the number of model features that are brought into correspondence with image features. Thus the task of model-based recognition can be viewed as finding legal transformations from a model to an image, and then determining whether one or more of these transformations accounts for a sufficiently large portion of the model and the observed data.

A number of recent model-based recognition systems have used affine transformations of the plane to represent the mapping from a two-dimensional model to a two-dimensional image (e.g. [4, 5, 12, 13, 16, 17, 18, 19, 21, 22]). This type of transformation can be used to approximate the two-dimensional image of a flat (planar) object at an arbitrary orientation in three-dimensional space. The transformation is equivalent to a three-dimensional rigid motion of the object, followed by orthographic projection and scaling (dilation). The scale factor accounts for the fact that objects which are farther away appear smaller than those which are close. This affine viewing model does not capture the perspective distortions that occur in real camera systems, because affine transformations preserve parallelism. It is a relatively good approximation to perspective except when an object is deep with respect to its distance from the viewer (e.g., railroad tracks going off to the horizon).

Recognition systems that make use of two-dimensional affine transformations fall into two basic classes. Methods in the first class explicitly compute an affine transformation based on the correspondence of a set of 'basis features' in the image and the model. This transformation is applied to the remaining model features in order to map them into the image coordinate frame, where they are compared with image features [2, 12, 13, 21]. Methods in the second class compute affine invariant representations of the model and the image, and directly compare these invariant representations [4, 5, 16, 17, 18, 19, 22]. In either case, recognition systems that employ affine transformations generally do not explicitly account for sensory uncertainty, but rather use some heuristic means to allow for uncertainty in the location of sensory data (one notable exception is [4] who formulate a probabilistic method). In this paper we provide a precise account of how uncertainty in the image measurements affects the range of transformations that are consistent with a given configuration of points acting under an affine transformation. This is important both for analyzing current recognition methods that employ affine transformations, and for developing new recognition methods that explicitly account for uncertainty.

We use our formal model of approximate affine matching to analyze two recognition methods that employ affine transformations: the geometric hashing method [16, 17, 18, 19, 22] and the alignment method [12, 13]. Under our model each sensor location is represented as an uncertainty disc of radius $\epsilon$, rather than as a specific $(x, y)$ location. For the alignment method, we use this model to provide a precise expression for the range of image point configurations that are consistent with a given quadruple of model points acting under an affine transformation. That is, we characterize what image points can match a given model point for a particular model and image basis (coordinate frame). This determines the range of possible matches for features, and hence the range of possible solutions to the recognition problem. For the geometric hashing method, we provide a similar analysis for the range of affine-invariant coordinates that are consistent with a given quadruple of points. This analysis reveals that when there is uncertainty in the data, the geometric hashing method cannot operate as originally proposed. The problem is that the uncertainty in the image point locations causes the range of values consistent with a given quadruple of model points to depend on the specific locations of the *image* points. The geometric hashing method proposes to build a fast lookup table based just on the model, and thus cannot account for the uncertainty using this table. We show how geometric hashing can be modified so that error can be precisely accounted for at run time, although this substantially changes the method.

## 1.1 Affine Transformations and Invariant Representations

An affine transformation of the plane can be represented as a nonsingular $2 \times 2$ matrix $\mathbf{L}$, and a 2-vector, $\mathbf{t}$, such that a given point $\mathbf{x}$ is transformed to $\mathbf{x}' = \mathbf{L}\mathbf{x} + \mathbf{t}$. It is well known that such a transformation maps any triple of points to any other triple (expect in degenerate cases), and that three points define an affine coordinate frame (analogous to a Cartesian coordinate frame in the case of Euclidean transformations) [6, 14]. In particular, a set of three points $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$ defines an affine coordinate frame in terms of which any other point $\mathbf{x}$ can be expressed using

$$\mathbf{x} = \mathbf{m}_1 + \alpha(\mathbf{m}_2 - \mathbf{m}_1) + \beta(\mathbf{m}_3 - \mathbf{m}_1). \qquad (1)$$

The values $\alpha$ and $\beta$ remain unchanged when a given affine transformation $A$ is applied to $\mathbf{x}$, $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$. That is,

$$A(\mathbf{x}) = A(\mathbf{m}_1) + \alpha(A(\mathbf{m}_2) - A(\mathbf{m}_1)) + \beta(A(\mathbf{m}_3) - A(\mathbf{m}_1)),$$

where $A$ is any affine transformation. Thus the pair $(\alpha, \beta)$ can be referred to as the affine-invariant coordinates of the point $\mathbf{x}$ with respect to the coordinate frame, or *basis*, $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$. We can think of $(\alpha, \beta)$ as a point in a two-dimensional space that we term the $\alpha$-$\beta$-plane.

The computation of an affine-invariant representation in terms of a coordinate frame $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$ has been used explicitly in the alignment [12, 13] and geometric hashing [16, 17, 18, 19, 22] methods. Both methods are motivated by the idea of finding sets of points in the image that are related to a corresponding set of points in the model by an affine transformation. The major difference between the two methods is in terms of whether the computations are done in a Euclidean space (i.e., the Cartesian coordinate systems of the model and the image) or the affine-invariant space of the $(\alpha,\beta)$ values. The alignment approach operates in the former domain, whereas the geometric hashing approach operates in the latter one.

We examine the effect of sensory uncertainty both in the Euclidean plane and the affine $\alpha$-$\beta$-space. In particular, we model each sensor point in terms of a *disc* of possible locations. The size of this disc is bounded by some given uncertainty factor, $\epsilon$. We then consider the range of values for a fourth point written in terms of the basis defined by the other three points, where all points have bounded uncertainty. We find that under this error model, in the Euclidean space the set of possible values for a given point $\mathbf{x}$ in terms of a basis $(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$ forms a disc whose radius depends on $\epsilon$, $\alpha$, and $\beta$. That is, assuming that each image point has a sensing uncertainty of magnitude $\epsilon$, the range of image locations that are consistent with $\mathbf{x}$ forms a circular region.

In the $\alpha$-$\beta$-space, the set of possible values of the affine coordinates of a point $\mathbf{x}$ in terms of a basis $(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$ forms an ellipse (except in degenerate cases). The area, center and orientation of this ellipse are given by somewhat complicated expressions that depend on the actual configuration of the basis points. The most important consequence of this analysis is that the set of possible values in the $\alpha$-$\beta$-plane *cannot* be computed independent of the actual locations of the basis points $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$, in the sensor coordinate system. In other words there is an interaction between the uncertainty in the sensor values and the actual locations of the sensor points. This limits the applicability of the geometric hashing method, as it requires that the $\alpha$-$\beta$ coordinates be computable independent of the actual location of the basis points (in order to construct a hash table offline).

Having derived expressions for the range of locations consistent with a given point $\mathbf{x}$ and a pair of bases $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$ and $(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$, we then use these expressions to analyze the sensitivity of the alignment and geometric hashing methods to the presence of sensor noise. We develop equations giving the probability that these methods will falsely report a match when none is present, using techniques similar to those developed in [9, 10]. For the geometric hashing method, our analysis assumes that the true elliptical regions in the $\alpha$-$\beta$-plane are being computed - even though the actual implementations of the geometric hashing method do not compute these values. Thus the real implementations will suffer even more from the problem of false matches (or alternatively will have the problem of missing correct matches).

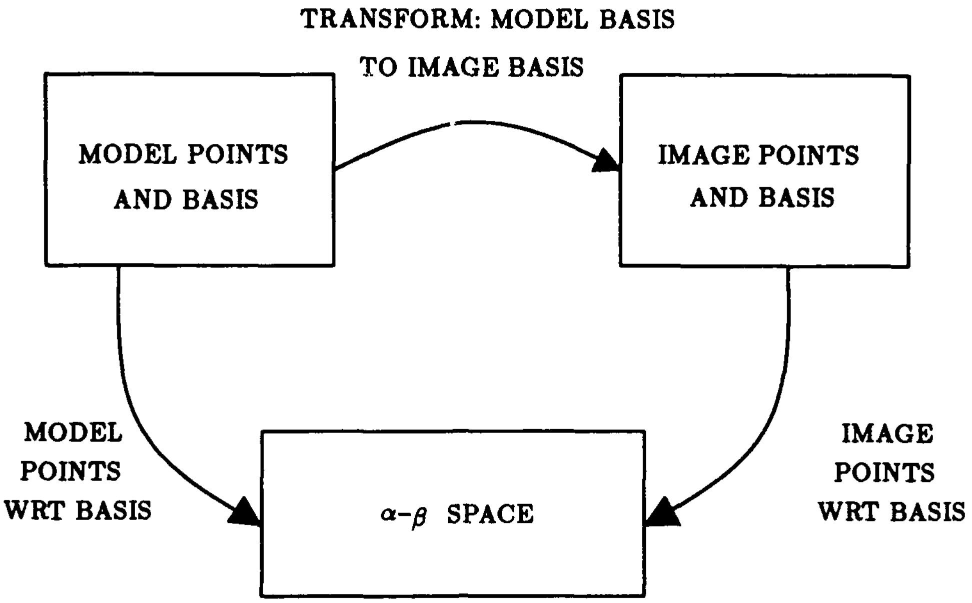

Figure 1: A schematization of the relation between the image coordinate frame, the model coordinate frame and the affine-invariant $\alpha$-$\beta$-space.

## 2 Image Uncertainty and Affine Coordinates

The main issue we wish to explore is the following: Given a model basis of three points and some additional model point, what sets of four image features are possible transformed instances of these points? In other words, for what quadruples of image points is there an affine transformation defined by pairing three of the image points with the three model basis points, such that the fourth model point is transformed into agreement with the fourth image point. Figure 1 schematizes the situation. A set of model points are given in a Cartesian coordinate frame, and some distinguished basis triple is also specified. Similarly a set of image points are given in their coordinate frame. Two different methods are used to map between the model and the image. One method, employed by geometric hashing, is to map both the model and the image points to $(\alpha, \beta)$ values using the basis triples. The other method, used by alignment, is to compute the transformation mapping the model basis to the image basis, and then use this transformation to map the model points to image coordinates. In both cases, a distinguished set of three model and image points is used to map a fourth point (or many such points) into some other space. We consider the effects that uncertainty has on these two methods, by modeling each image point as an $\epsilon$-sized disc of possible locations rather than as a specific point.

First we characterize the range of image measurements in the $x$-$y$ (Euclidean) plane that are consistent with the $(\alpha, \beta)$ pair computed for a given quadruple of model points, as specified by equation (1). This corresponds to the case of explicitly computing a transformation from one Cartesian coordinate frame (the model) to another (the image). We find that if the uncertainty in the locations of the sensor points is bounded by a disc of radius $\epsilon$, then the range of possible image measures consistent with a given $(\alpha, \beta)$ pair is a disc with radius bounded below by $\epsilon(1 + |\alpha| + |\beta|)$ and above by $2\epsilon(1 + |\alpha| + |\beta|)$. This defines the set of image points that could match a specific model point, given both an image and model basis.

We then perform the same analysis for the range of affine coordinate, or $\alpha$ and $\beta$, values that are consistent with a given quadruple of points. This corresponds to the case of mapping both the model and image points to $(\alpha, \beta)$ values. In order to do this, we use the expressions that we derived for the Euclidean case to determine the region of $\alpha$-$\beta$-space that is consistent with a given point and basis. This region of $\alpha$-$\beta$-space is in general an ellipse containing the point $(\alpha, \beta)$ (but not necessarily centered at that point). The expressions for the size, orientation and position of the ellipse depend on the actual locations of the points defining the basis.

Assume that we are given three model points, $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$, and the affine coordinates $(\alpha, \beta)$ of a fourth model point $\mathbf{x}$ defined by

$$\mathbf{x} = \mathbf{m}_1 + \alpha(\mathbf{m}_2 - \mathbf{m}_1) + \beta(\mathbf{m}_3 - \mathbf{m}_1). \tag{2}$$

Further assume that we are given three sensor points $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$, such that

$$\mathbf{s}_i = T(\mathbf{m}_i) + \mathbf{e}_i,$$

where $T$ is some affine transformation, and $\mathbf{e}_i$ is an arbitrary vector of magnitude at most $\epsilon_i$. That is, $T$ is some underlying affine transformation that cannot be directly observed in the data because each data point has been perturbed by some arbitrary vector $\mathbf{e}_i$. These error vectors $\mathbf{e}_i$ are assumed to be bounded by $\epsilon_i$, using our error model that represents a point as a disc of radius $\epsilon_i$. (Note that in general we will always use $\epsilon_i = \epsilon$, but in principle one could allow different amounts of bounded uncertainty with different features.)

We are interested in the possible locations of a fourth sensor point, call it $\hat{\mathbf{x}}$, such that $\hat{\mathbf{x}}$ could correspond to the ideally transformed point $T(\mathbf{x})$. We note that the possible positions of $\hat{\mathbf{x}}$ are affected both by the sensor error in measuring each image basis point, $\mathbf{s}_i$, and by the error in measuring the fourth point itself. Thus the possible locations are given by transforming the invariant representation of equation (2) and adding in the error $\mathbf{e}_0$ from measuring $\mathbf{x}$,

$$\begin{aligned}
\hat{\mathbf{x}} &= T(\mathbf{m}_1 + \alpha(\mathbf{m}_2 - \mathbf{m}_1) + \beta(\mathbf{m}_3 - \mathbf{m}_1)) + \mathbf{e}_0 \\
&= \mathbf{s}_1 - \mathbf{e}_1 + \alpha(\mathbf{s}_2 - \mathbf{e}_2 - \mathbf{s}_1 + \mathbf{e}_1) + \beta(\mathbf{s}_3 - \mathbf{e}_3 - \mathbf{s}_1 + \mathbf{e}_1) + \mathbf{e}_0 \\
&= \mathbf{s}_1 + \alpha(\mathbf{s}_2 - \mathbf{s}_1) + \beta(\mathbf{s}_3 - \mathbf{s}_1) - \mathbf{e}_1 + \alpha(\mathbf{e}_1 - \mathbf{e}_2) + \beta(\mathbf{e}_1 - \mathbf{e}_3) + \mathbf{e}_0.
\end{aligned}$$

That is, the measured point $\hat{\mathbf{x}}$ can lie in a range of locations about the ideal location, specified by

$$\mathbf{s}_1 + \alpha(\mathbf{s}_2 - \mathbf{s}_1) + \beta(\mathbf{s}_3 - \mathbf{s}_1).$$

This range of possible locations is specified by the linear combination of the four error vectors

$$-\mathbf{e}_1 + \alpha(\mathbf{e}_1 - \mathbf{e}_2) + \beta(\mathbf{e}_1 - \mathbf{e}_3) + \mathbf{e}_0,$$

or equivalently

$$-\left[(1-\alpha-\beta)\mathbf{e}_1 + \alpha\mathbf{e}_2 + \beta\mathbf{e}_3 - \mathbf{e}_0\right], \tag{3}$$

where each $\mathbf{e}_i$ is an arbitrary vector of length at most $\epsilon_i$.

The set of all possible locations specified by a given $\mathbf{e}_i$ is a disc of radius $\epsilon_i$ about the origin, which we denote $C(\epsilon_i)$:

$$C(\epsilon_i) = \{\mathbf{e}_i \;\mid\; \|\mathbf{e}_i\| \leq \epsilon_i\}.$$

Similarly, the product of any constant $k$ with $\mathbf{e}_i$ yields a disc $C(k\epsilon_i)$ of radius $|k|\epsilon_i$ centered about the origin. Thus substituting the expressions for the disc in equation (3) we obtain the following expression for the set of all locations about the ideal point $\mathbf{s}_1 + \alpha(\mathbf{s}_2 - \mathbf{s}_1) + \beta(\mathbf{s}_3 - \mathbf{s}_1)$:

$$C([1-\alpha-\beta]\epsilon_1) \oplus C(\alpha\epsilon_2) \oplus C(\beta\epsilon_3) \ominus C(\epsilon_0), \tag{4}$$

where $\oplus$ denotes the Minkowski sum of sets. That is, given two sets $A$ and $B$, $A \oplus B = \{p + q | p \in A, q \in B\}$ (and similarly for $\ominus$).

In order to simplify the expression for the range of $\hat{\mathbf{x}}$ we make use of the following fact, which follows directly from the definition of the Minkowski sum for sets.

**Claim 1** $C(r_1) \oplus C(r_2) = C(r_1) \ominus C(r_2) = C(r_1 + r_2)$, *where* $C(r_i)$ *is a disc of radius* $r_i$ *centered about the origin,* $r_i \geq 0$.

If we assume that the $\epsilon_i$ are all equal to $\epsilon$ (i.e., all the sensor error bounds are the same), then using Claim 1 we can simplify equation (4) to

$$C\left(\epsilon[|1-\alpha-\beta| + |\alpha| + |\beta| + 1]\right).$$

The absolute values arise from the fact that $\alpha$ and $\beta$ can become negative, but the radius of a disc is a positive quantity. Clearly the radius of the error disc grows with increasing magnitude of $\alpha$ and $\beta$, but the actual expression governing this growth is different for different portions of the $\alpha-\beta$-plane, as shown in Figure 2 (the diagonal line in the figure is $1 - \alpha - \beta = 0$). In particular, the absolute values will lead to different expressions for the radius of the error disc as a function of $\alpha$ and $\beta$, as illustrated in the figure.

We can bound the expressions defining the radius of the uncertainty disc by noting that

$$1 + |\alpha| + |\beta| \leq (|1-\alpha-\beta| + |\alpha| + |\beta| + 1) \leq 2(1 + |\alpha| + |\beta|).$$

We have thus established the following result:

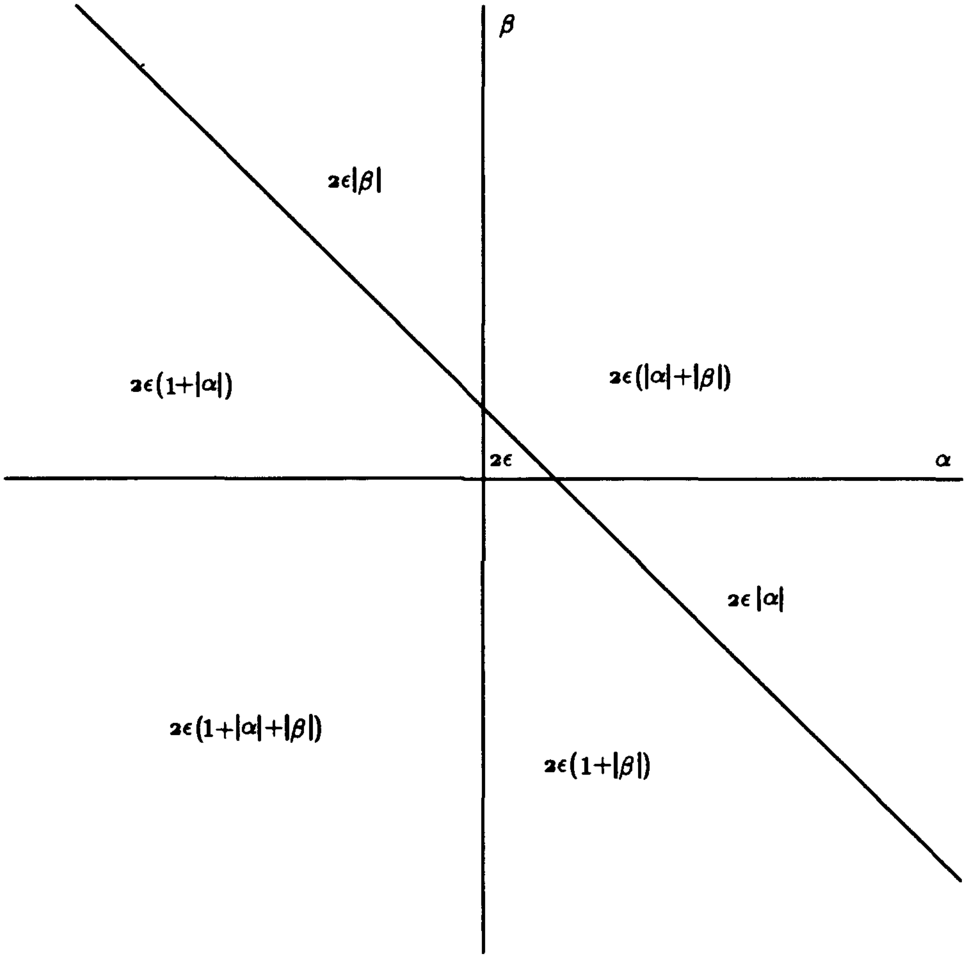

Figure 2:

Diagram of error effects. The region of feasible points is a disc, whose radius is given by the indicated expression, depending on the values of $\alpha$ and $\beta$.

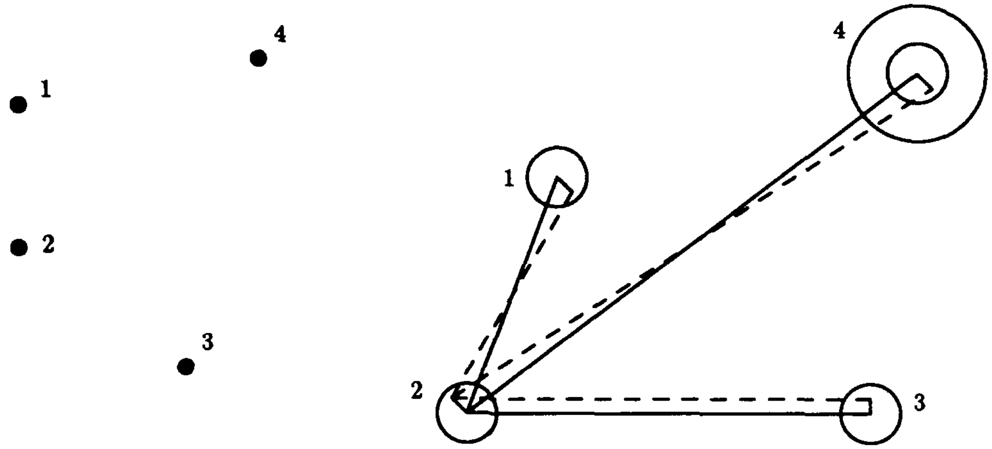

Figure 3:
Diagram of error effects. A set of four model points are shown on the left. The positions of four image points are shown on the right, three of which are used to establish a basis. The actual position of each transfcrmed model point corresponding to the basis image points is offset by an error vector of bounded magnitude. The coordinates of the fourth point, written in terms of the basis vectors, can thus vary from the ideal case, shown in solid lines, to cases such as that shown in dashed lines. This leads to a disc of variable size in which the corresponding fourth model point could lie.

**Proposition 1** *The range of image locations that is consistent with a given pair of affine coordinates $(\alpha, \beta)$ is a disc of radius $r$, where*

$$\epsilon(1 + |\alpha| + |\beta|) \leq r \leq 2\epsilon(1 + |\alpha| + |\beta|)$$

*and where $\epsilon$ is a positive constant that bounds the positional uncertainty of the image data.*

The effect of this circular uncertainty region for the location of $\hat{\mathbf{x}}$ is illustrated in Figure 3. The positional uncertainty in the locations of the three image basis points results in a circle of possible locations for the fourth point. The error in measuring the fourth point itself increases the radius of this error disc.

The expression in Proposition 1 allows the calculation of error bounds for any method based on two-dimensional affine transformations, such as [2, 12, 21]. In particular, if $|\alpha|$ and $|\beta|$ are both less than 1, then the error in the position of a point is at most $6\epsilon$. This condition can be met by using as the affine basis, three points $\mathbf{m}_1, \mathbf{m}_2$ and $\mathbf{m}_3$ that lie on the convex hull of the set of model points, and are maximally separated from one another.

It should be noted that the expression in Proposition 1 is independent of the actual locations of the model or image points. This means that the possible positions of the fourth point vary only with the sensor error and the values of $\alpha$ and $\beta$. They do not vary with the configuration of the model basis (e.g., even if close to collinear) nor do they vary with the configuration of the image basis. In other words, the error range does not depend on the viewing direction. Even if the model is viewed end on, so that all three model points appear nearly co-linear, or if the model is viewed at a small scale, so that all three model points are close together, the size of the region of possible locations of the fourth model point in the image will remain unchanged.

The viewing direction does, however, greatly affect the affine coordinate system defined by the three projected model points. Thus the set of possible *affine coordinates* of the fourth point, when considered directly in $\alpha$-$\beta$-space, will vary greatly. Our next goal is to characterize this set of affine coordinates. This can be done by making use of Proposition 1, which tells us the set of image locations consistent with a fourth point. Implicit in this analysis is the set of affine transformations that produce possible fourth image point locations. This can in turn be used to characterize the range of $(\alpha, \beta)$ values that are consistent with a given set of four points.

We will do the analysis using the upper bound on the radius of the error disc from Proposition 1. In actuality, the analysis is slightly more complicated, because the expression governing the disc radius varies as shown in Figure 2. For our purposes, however, considering the extreme case is sufficient. It should also be noted from the figure that the extreme case is in fact quite close to the actual value over much of the range of $\alpha$ and $\beta$.

Given a triple of image points that form a basis, and a fourth image point, $\mathbf{s}_4$, we are interested in determining the range of affine coordinates for the fourth point that are consistent with the possibly erroneous image measurements. In effect, each sensor point $\mathbf{s}_i$ takes on a range of possible values, and each quadruple of such values produces a possibly distinct value using equation (1). As illustrated in Figure 4 we could determine all the feasible values by varying the basis vectors over the uncertainty discs associated with their endpoints, finding the set of $(\alpha', \beta')$ values such that the resulting point in this affine basis lies within $\epsilon$ of the original point. By our previous results, however, it is equivalent to find affine coordinates $(\alpha', \beta')$ such the Euclidean distance from

$$\mathbf{s}_1 + \alpha'(\mathbf{s}_2 - \mathbf{s}_1) + \beta'(\mathbf{s}_3 - \mathbf{s}_1)$$

to

$$\mathbf{s}_1 + \alpha(\mathbf{s}_2 - \mathbf{s}_1) + \beta(\mathbf{s}_3 - \mathbf{s}_1)$$

is bounded above by $2\epsilon(1 + |\alpha'| + |\beta'|)$.

The boundary of the region of such points $(\alpha', \beta')$ is defined by requiring the distance from the nominal image point

$$\mathbf{s}_4 = \mathbf{s}_1 + \alpha(\mathbf{s}_2 - \mathbf{s}_1) + \beta(\mathbf{s}_3 - \mathbf{s}_1)$$

to be $2\epsilon(1 + |\alpha'| + |\beta'|)$, which is when

$$[2\epsilon(1 + |\alpha'| + |\beta'|)]^2 = [(\alpha - \alpha')u]^2 + 2(\beta - \beta')(\alpha - \alpha')vu\cos\phi + [(\beta - \beta')v]^2 \quad (5)$$

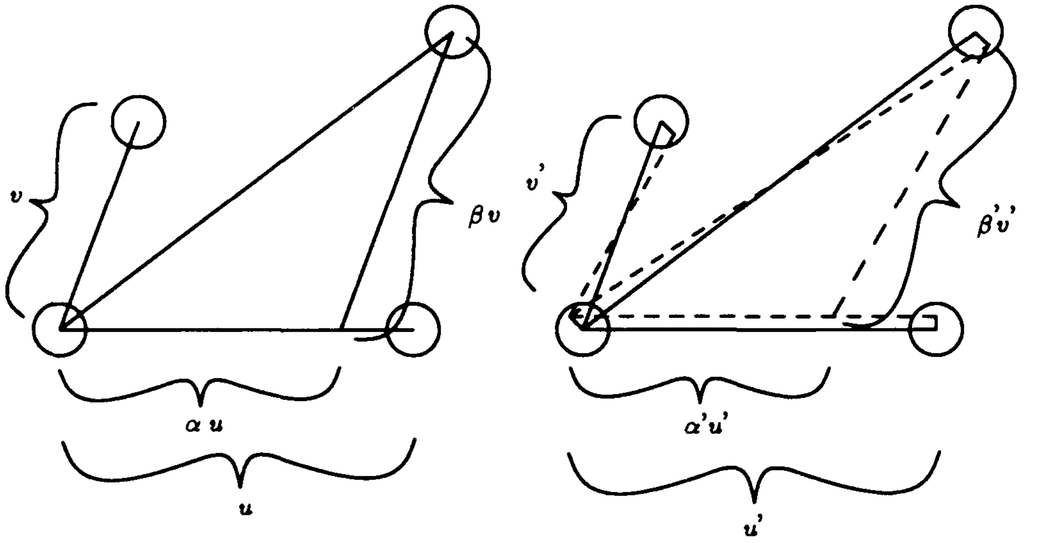

Figure 4:

The example on the left shows the canonical example of affine coordinates. The fourth point is offset from the origin point by the sum of $\alpha$ times the first basis vector **u** plus $\beta$ times the second basis vector **v**. The example on the right shows a second consistert set of affine coordinates. By taking other vectors that lie within the uncertainty regions of each of the image points, we can find a different set of affine coordinates $\alpha', \beta'$ such that the new fourth point based on these coordinates also lies within the uncertainty bound of the image point.

where

$$\begin{aligned} \mathbf{u} &= \mathbf{s}_2 - \mathbf{s}_1 \\ \mathbf{v} &= \mathbf{s}_3 - \mathbf{s}_1 \\ u &= \|\mathbf{u}\| \\ v &= \|\mathbf{v}\| \end{aligned}$$

and where the angle made by the image basis vectors $\mathbf{s}_2 - \mathbf{s}_1$ and $\mathbf{s}_3 - \mathbf{s}_1$ is $\phi$. Considered as an implicit function of $\alpha', \beta'$, equation (5) defines a conic. If we expand out equation (5), we get

$$a_{11}(\alpha')^2 + 2a_{12}\alpha'\beta' + a_{22}(\beta')^2 + 2a_{13}\alpha' + 2a_{23}\beta' + a_{33} = 0 \tag{6}$$

where

$$\begin{aligned} a_{11} &= u^2 - 4\epsilon^2 \\ a_{22} &= v^2 - 4\epsilon^2 \\ a_{12} &= vu\cos\phi - 4s_\alpha s_\beta \epsilon^2 \\ a_{13} &= -u\left[\alpha u + \beta v \cos\phi\right] - 4s_\alpha \epsilon^2 \\ a_{23} &= -v\left[\alpha u \cos\phi + \beta v\right] - 4s_\beta \epsilon^2 \\ a_{33} &= \alpha^2 u^2 + 2\alpha\beta uv\cos\phi + \beta^2 v^2 - 4\epsilon^2 \end{aligned}$$

and where

$$\begin{aligned} s_\alpha &= \begin{cases} 1 & \text{if } \alpha' \geq 0, \\ -1 & \text{if } \alpha' < 0, \end{cases} \\ s_\beta &= \begin{cases} 1 & \text{if } \beta' \geq 0, \\ -1 & \text{if } \beta' < 0. \end{cases} \end{aligned}$$

For notational simplicity in what follows, it is convenient to assume that $\alpha$ and $\beta$ are positive, so that $s_\alpha = 1, s_\beta = 1$.

We can use this form to compute the invariant characteristics of a conic [15]:

$$I = u^2 + v^2 - 8\epsilon^2 \tag{7}$$

$$D = u^2v^2\sin^2\phi - 4\epsilon^2\left(u^2 - 2uvs_\alpha s_\beta \cos\phi + v^2\right) \tag{8}$$

$$A = -4\epsilon^2 u^2 v^2 \sin^2\phi(1 + s_\alpha\alpha + s_\beta\beta)^2 \tag{9}$$

If $u^2 + v^2 > 8\epsilon^2$, then $\frac{A}{I} < 0$. Furthermore, if

$$u^2v^2\sin^2\phi > 4\epsilon^2\left(u^2 - 2uvs_\alpha s_\beta\cos\phi + v^2\right)$$

then $D > 0$ and the conic defined by equation (5) is an ellipse. We will ignore the degenerate cases in which the conic is not an ellipse. Such cases only occur either when the image basis points are very close together, or when the image basis points are nearly collinear. For instance, as long as the image basis vectors **u** and **v** are each at least $2\epsilon$ in length then $u^2 + v^2 > 8\epsilon^2$. Similarly, as long as $\sin\phi$ is not small, $D > 0$. In fact, cases where these conditions do not hold will be very unstable and thus should be avoided anyway.

Given the conic invariants, we can compute a number of characteristics of the ellipse. The area of the ellipse is given by

$$\frac{4\pi\epsilon^2 u^2 v^2 \sin^2\phi(1 + s_\alpha\alpha + s_\beta\beta)^2}{[u^2 v^2 \sin^s\phi - 4\epsilon^2\,(u^2 - 2uvs_\alpha s_\beta\cos\phi + v^2)]^{\frac{3}{2}}}. \tag{10}$$

The center of the ellipse is at

$$\alpha_0 = \frac{1}{D}\left[\alpha u^2 v^2 \sin^2\phi - 4\epsilon^2(\alpha u^2 - s_\alpha(1 + s_\beta\beta)v^2 + uv\cos\phi(\beta + s_\beta(1 - s_\alpha\alpha)))\right]$$

$$\beta_0 = \frac{1}{D}\left[\beta u^2 v^2 \sin^2\phi - 4\epsilon^2(\beta v^2 - s_\beta(1 + s_\alpha\alpha)u^2 + uv\cos\phi(\alpha + s_\alpha(1 - s_\beta\beta)))\right]. \tag{11}$$

The angle of the principal axes, $\Phi$, with respect to the $\alpha$ axis is

$$\tan 2\Phi = \frac{2[uv\cos\phi - 4\epsilon^2 s_\alpha s_\beta]}{u^2 - v^2}. \tag{12}$$

Thus we have established the following:

**Proposition 2** *Given bounded errors of $\epsilon$ in the measurement of the image points, the region of uncertainty associated with a pair of affine coordinates $(\alpha, \beta)$ in $\alpha$-$\beta$-space is an ellipse. The area of this ellipse is given by equation (10), the center is at $(\alpha_0, \beta_0)$ as given by equation (11), and the orientation is given by equation (12).*

In other words, given a set of four points whose locations are only known to within discs of radius $\epsilon$, there is an ellipse-shaped region of possible $(\alpha, \beta)$ values specifying the location of one point with respect to the other three. Thus if we compare $(\alpha, \beta)$ values generated by some model of an object with those specified by an image, when there is $\epsilon$-uncertainty in the image data, each image datum actually specifies an ellipse of $(\alpha, \beta)$ values. The area of this ellipse depends on the degree of sensor uncertainty, $\epsilon$, the values of $\alpha$ and $\beta$, and the configuration of the three image points that form the basis. In order to compare the model values with image values it is necessary to check that the affine-invariant coordinates for each model point lie within the elliptical region of possible affine-invariant values associated with the corresponding image point.

The fact that the regions of consistent parameters in $\alpha$-$\beta$-space are ellipses causes some difficulties for discrete hashing schemes, such as the one employed by geometric hashing. This is discussed in greater detail in a later section, but the basic idea of the geometric hashing method is to compute affine coordinates of model points with respect to some choice of basis, and to use these affine coordinates as the hash keys to store the basis in a table. In general, the implementations of this method use square buckets to tessellate the hash space (the $\alpha$-$\beta$-space). In this case, we see that even if we chose buckets whose size is commensurate with the ellipse, several such buckets are likely to intersect any given ellipse due to the difference in shape of the two regions. Thus, it is necessary to hash to multiple buckets, and this increases the probability that a random pairing of model and image bases will receive a significant number of votes.

A further problem for discrete hashing schemes is the fact that the size of the ellipse increases as a function of $(1+|\alpha|+|\beta|)^2$. Thus points with larger affine coordinates give rise to larger ellipses than those with smaller coordinates. The contours along which the centers of equal-sized ellipses lie are parabolic arcs (i.e. contours of constant $1+|\alpha|+|\beta|$), rather than circles. Either one must hash a given value to many buckets, or one must account for this effect by sampling the space in a manner that varies with parabolic distance, but this would require some careful analysis.

The most critical issue for discrete hashing schemes is the fact that the shape, orientation and position of the ellipse depends on the specific image basis chosen. That is, the orientation of the ellipse changes as $u, v$ and $\phi$ change (which are parameters computed from the image basis). This means that there is no clear way to fill the hash table as a pre-processing step, independent of a given image, which is a crucial part of the geometric hashing method. The problem is that the error ellipse associated with a given $(\alpha, \beta)$ pair depends on the characteristics of the image basis, and we don't know that ahead of time. There is no way to pre-compute these error regions because they depend inherently on the image point configuration. This means it is either necessary to approximate the ellipses by assuming bounds on the possible image basis, which will allow both false positive and false negative hits in the hash table, or to compute the ellipse to access at run time. Note that the geometric hashing method does not address any of these issues. It is simply assumed that some 'appropriate' tessellation of the image space exists.

In summary, in this section we have characterized the range of image coordinates and the range of $(\alpha, \beta)$ values that are consistent with a given point, with respect to some basis, when there is uncertainty in the image data. In the following section we analyze what fraction of all possible points (in some bounded image region) are consistent with a given range of $(\alpha, \beta)$ values. Then in the subsequent sections we use this to derive expressions for the probability of a false match for both the geometric hashing method and the alignment method.

# 3 The Selectivity of Affine-Invariant Representations

We are interested in determining the probability than an object recognition system will erroneously report an instance of an object in an image. Recall that such an instance in general is specified by giving a transformation from model coordinates to image coordinates, and a measure of 'quality' based on the number of model features that are paired with image features under this transformation. Thus we are interested in whether a random association of model and image features can occur in sufficient number to masquerade as a correct solution. We use the results developed above in order to determine the probability of such a *false match.* There are two stages to this analysis; the first is a statistical analysis that is independent of the given recognition method, and the second is a combinatorial analysis that depends on the particular recognition method. In this section we examine the first stage, and then in subsequent sections we turn to the analysis of the geometric hashing and alignment methods.

In order to determine the probability that a match will be falsely reported we need to know the 'selectivity' of a quadruple of model points. Recall from Figure 1 that each model point is mapped to a point $\alpha$-$\beta$-space with respect to a particular model basis (triple). Similarly each image point, modeled as a disc, is mapped to an elliptical region of possible points in $\alpha$-$\beta$-space. Each such image region that contains one or more model points specifies an image point that is consistent with the given model. Thus we need to estimate the probability that a given image basis and fourth image point chosen at random will map to a region of $\alpha$-$\beta$-space that is consistent with one of the model points written in terms of some model basis. One way of characterizing this is in terms of the proportion of the $\alpha$-$\beta$-space that is consistent with a given basis and fourth point (where the size of the space is bounded in some way). As was shown above, the elliptical regions in $\alpha$-$\beta$-space are equivalent to circular regions in image space. Thus, for ease of analysis we choose to work with the formulation in terms of circles in image space.

To determine the selectivity, we assume we are given some image basis and a potential corresponding model basis. Each of the remaining $m - 3$ model points are defined as affine coordinates relative to the model basis. These can then be transformed into the image domain, by using the same affine coordinates, with respect to the image basis. Because of the uncertainty of the image points, there is an uncertainty in the associated affine transformation. This manifests itself as a range of possible positions for the model points, as they are transformed into the image. Previously we determined that a transformed model point had to be within $2\epsilon(1 + |\alpha| + |\beta|)$ of an image point in order to match it. That calculation took into account error in the matched image point as well as the basis image points. Therefore, placing an appropriately sized disc about each model point is equivalent to placing an $\epsilon$ sized disc about each image point. We thus represent each transformed model point as giving rise to a disc of some radius, positioned relative to the nominal position of the model point with respect to the image basis. For convenience, we use the upper bound on the size of the radius, $2\epsilon(1 + |\alpha| + |\beta|)$. For each model point, rewritten in image coordinates, we need to know the probability that at least one image point lies in the associated error disc about the transformed model point, because if this happens it means that there is a consistent model and image point for the given model and image basis. To estimate this probability, we need to estimate the expected size of the disc. Since the disc size varies with $|\alpha| + |\beta|$, this means we need an estimate of the distribution of points with respect

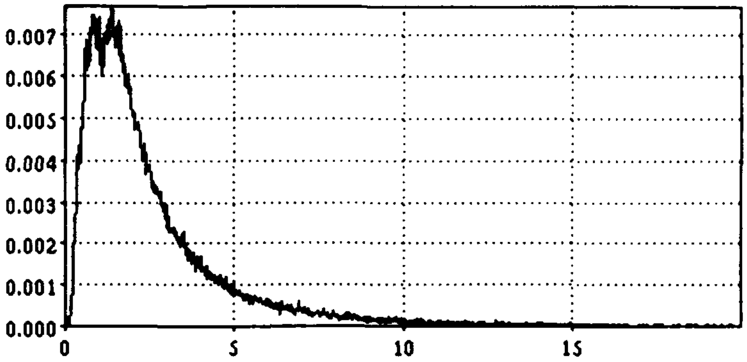

Figure 5:
Histogram of distribution of $|\alpha| + |\beta|$ values. Vertical axis is ratio of number of samples to total samples, horizontal axis is value for $|\alpha| + |\beta|$. The maximum over 300,000 samples was 51. Only the first portion of the graph is displayed.

to affine coordinates. In fact, by Figure 2 we should find the distribution of points as a function of $(\alpha, \beta)$ since the disc sizes varies with these values. This is messy, and thus we use an approximation instead.

For this approximation, we measure the distribution with respect to $\rho$, where $\rho = |\alpha| + |\beta|$, since both the upper and lower bounds on the disc size are functions of this variable. Intuitively we expect the distribution to vary inversely with $\rho$. To verify this, we ran the following experiment. A set of 25 points were generated at random, with the property that their pairwise minimum separation was at least 25 pixels, and their pairwise maximum separation was at most 250 pixels. All possible bases were selected, and for each basis for which the angle between the axes was at least $\pi/16$, all the other model points were rewritten in terms of affine invariant coordinates $(\alpha, \beta)$. This gave roughly 300,000 samples, which we histogrammed with respect to $\rho(\alpha, \beta) = |\alpha| + |\beta|$. We found that the maximum value for $\rho$ in this case was roughly 51. In general, however, almost all of the values were much smaller, and indeed, the distribution showed a strong inverse drop off, as can be seen from Figure (5).

Given this evidence, we considered two different models for the distribution of points in affine coordinates. The first is:

$$\delta(\alpha, \beta) = \begin{cases} k\rho & \rho \leq 1 \\ \frac{k}{\rho^2} & \rho \geq 1. \end{cases} \tag{13}$$

Figure (6) illustrates the fit of this to the actual data. The second is:

$$\delta(\alpha, \beta) = \begin{cases} k\rho & \rho \leq 1 \\ \frac{k}{\rho^{\frac{3}{2}}} & \rho \geq 1. \end{cases} \tag{14}$$

Figure (7) illustrates the fit of this to the actual data.

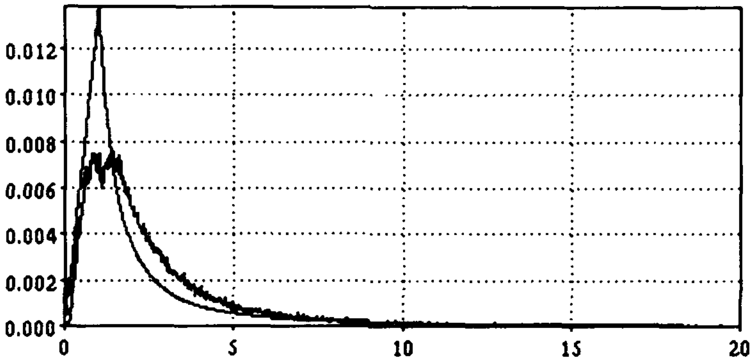

Figure 6:
Histogram of distribution of $|\alpha| + |\beta|$ values. Vertical axis is ratio of number of samples to total samples, horizontal axis is value for $|\alpha| + |\beta|$. The maximum over 300,000 samples was 51. Only the first portion of the graph is displayed. Overlayed with this is a $\rho^{-2}$ distribution.

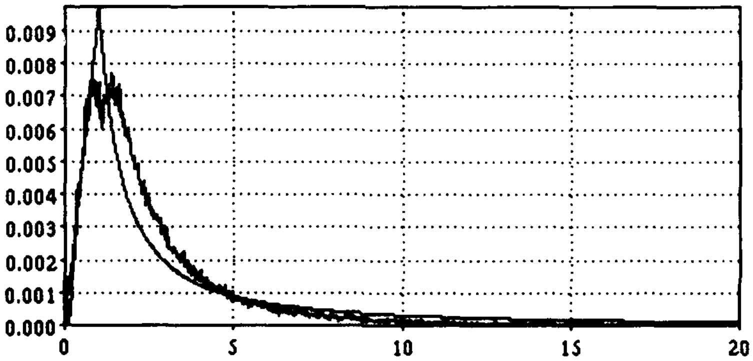

Figure 7: Histogram of distribution of $|\alpha| + |\beta|$ values. Vertical axis is number of samples, horizontal axis value for $\rho = |\alpha| + |\beta|$. The maximum over 300,000 samples was 51. Only the first portion of the graph is displayed. Overlayed with this is a $\rho^{-\frac{3}{2}}$ distribution.

We choose to use the first model, because it underestimates the probability for large values of $\rho$, at the cost of overestimating it for small values of $\rho$. Since we are interested in finding the expected size of the error disc, and this grows with $\rho$ such an approximation will underestimate the size of the disc.

First, we integrate equation (13) over all possible values and normalize to 1 in order to deduce the constant:

$$k = \frac{2}{3 - \frac{2}{\rho_m}} \tag{15}$$

where $\rho_m$ is the maximum value for $\rho$ (and $\rho = |\alpha| + |\beta|$).

Next, we want to find the expected area of a disc in image space. Recall that we are going to examine the upper bound on the disc size, so that in principle, this area is just

$$4\pi\epsilon^2(1+\rho)^2.$$

We could simply integrate this with respect to the distribution from equation (13)

$$\int_{\rho=0}^{\rho_m} 4\pi\epsilon^2(1+\rho)^2\delta(\rho)d\rho.$$

This, however, ignores the fact that the image is of finite size (say each dimension is $2r$), and some of the disc may lie beyond the bounds of the image. We therefore separate out four different cases.

The first case is for $\rho \leq 1$. Here we get

$$A_1 = \int_{\rho=0}^{1} 4\pi\epsilon^2(1+\rho)^2 k\rho\, d\rho = 4\pi\epsilon^2 k\frac{17}{12}. \tag{16}$$

The second case considers discs that will lie entirely within the bounds of the image. Consider figure 8, which shows the limiting case, assuming that the coordinate frame of the basis is centered at the center of the image, and the image dimensions are $2r$ by $2r$. In this case, we have

$$r - p \geq \gamma$$

where $\gamma = 2\epsilon(1+\rho)$. In general, we have $p \leq \rho d$ where $d$ is the separation between two of the basis points in the image, and this leads to the condition that if $1 \leq \rho \leq c_1$ where

$$c_1 = \min\left\{\rho_m, \frac{r-2\epsilon}{2\epsilon+d}\right\}$$

then the discs will all lie entirely within the image. Thus the second case is

$$\begin{aligned} A_2 &= \int_{\rho=1}^{c_1} 4\pi\epsilon^2(1+\rho)^2 k\rho^{-2}\, d\rho \\ &= 4\pi\epsilon^2 k\left[c_1 + 2\log c_1 - \frac{1}{c_1}\right] \\ &= 4\pi\epsilon^2 k\left[2\log\left(\frac{r-2\epsilon}{d+2\epsilon}\right) + \frac{r^2 - d^2 - 4\epsilon(r+d)}{(d+2\epsilon)(r-2\epsilon)}\right]. \end{aligned} \tag{17}$$

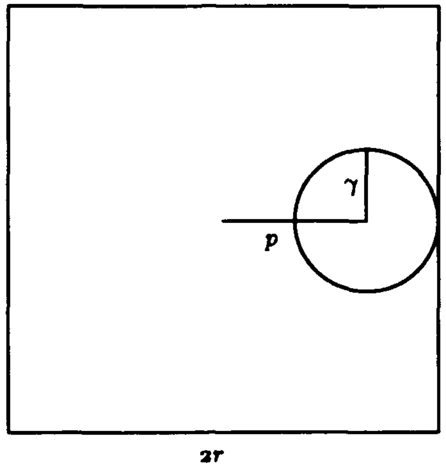

Figure 8:
Case 2. Limiting case of an error disc lying entirely within the image, assuming the coordinate basis is center in the image.

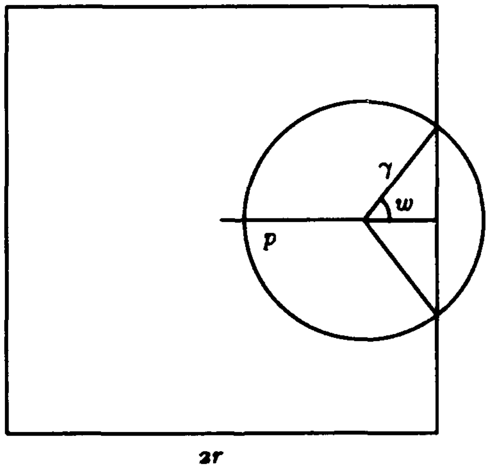

Figure 9:
Case 3. Example of case in which the image error disc does not lie entirely within the bounds of the image.

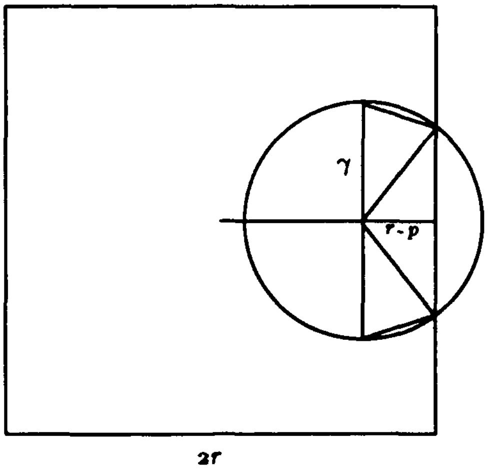

Figure 10:
Case 3. Underestimate of the area of the image error disc.

The final expansion is based on the assumption that $\rho_m > c_1$, which is true for virtually all cases of interest.

In the third case, when $\rho_m > c_1$, for values of $\rho > c_1$ there is some truncation of the disc. This situation is shown in figure 9. In this case the area of the portion of the disc lying inside the image is given by

$$\gamma^2 \left[\pi - \cos^{-1}\left(\frac{r-p}{\gamma}\right)\right] + (r-p)\left[\gamma^2 - (r-p)^2\right]^{\frac{1}{2}}. \qquad (18)$$

Integrating this with respect to the distribution $\delta(\rho)$ is messy. Because we are interested in underestimating the expected area of the discs, we can use the following approximation. For values of $p$ ranging from $c_1$ to $c_2$, where $c_2$ is the value for which $\rho d$ reaches the edge of the image, we can underestimate the area of the disc contained within the image, by using the faceted approximation shown in Figure 10. The actual expression for the area in the third case, $A_3$, is relatively complex, and is given in Appendix A.

The final case occurs when the actual point is beyond the limits of the image, but the disc size is large enough that some portion of it intersects the image. The case is shown in Figure 11, as well as the approximation we use to underestimate the area. Again the expression is complex, and is given in Appendix A.

Depending on the specific values for $\rho_m$, $c_1$ and $c_2$ we can add in the appropriate contributions from equations 16, 17, 32 and 34, together with the value for $k$ (from equation (15)) to obtain an underestimate for the expected area of an error disc — the expected area of a circle in image space that will be consistent with a point expressed in terms of some affine basis. Since such discs can in general occur with equal probability anywhere in the image, the probability that a model point lies within a disc associated with an image point is simply the ratio of this area to the

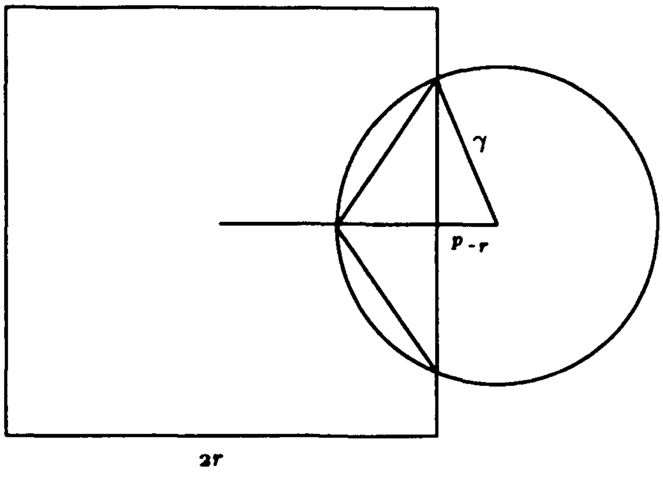

Figure 11:
Case 4. Underestimate of the area of the image error disc.

area of the image. Thus by normalizing these equations, by dividing by $(2r)^2$, where $r$ is half the diameter of the image, we have an underestimate for the selectivity of the scheme.

This leads to the following estimate for the selectivity of the scheme:

**Proposition 3** *Given a model basis and a fourth model point, the probability that an image basis and a fourth image point, hypothesized to correspond to the model basis and point, will map at random to a region of $\alpha$-$\beta$-space consistent with the model point and basis is given by*

$$\mu = \frac{A_1 + A_2 + A_3 + A_4}{4r^2} \tag{19}$$

*where the $A_i$'s are given by equations 16, 17, 32, and 34.*

This is based on using the upper bound on the radius of the error discs. As noted earlier, a simple lower bound can be obtained by substituting $\epsilon/2$ in place of $\epsilon$, reflecting the use of the bound $\epsilon(1+\rho)$ in place of $2\epsilon(1+\rho)$. In this case, the bounds $c_1$ and $c_2$ will change slightly.

We can use this to compute example values for the selectivity, which depends on $\rho_m$ (the maximum value of $|\alpha| + |\beta|$). If we allow any possible triple of points to form a basis, then $\rho_m$ can be arbitrarily large. Consider the example shown in Figure (12). The value for $\rho$ associated with the point $p$ is given by

$$\frac{p}{uv|\sin\phi|}\left(u|\sin\theta| + v|\sin(\phi-\theta)|\right).$$

As $\phi$ approaches 0, this value becomes unbounded. We can exclude unstable bases if we set limits on the allowable range of values for $\phi$, in particular, we can restrict

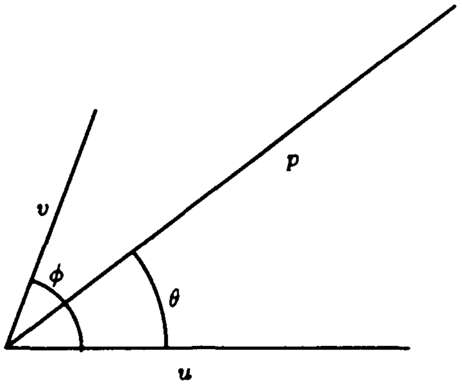

Figure 12:
Diagram of affine coordinates.

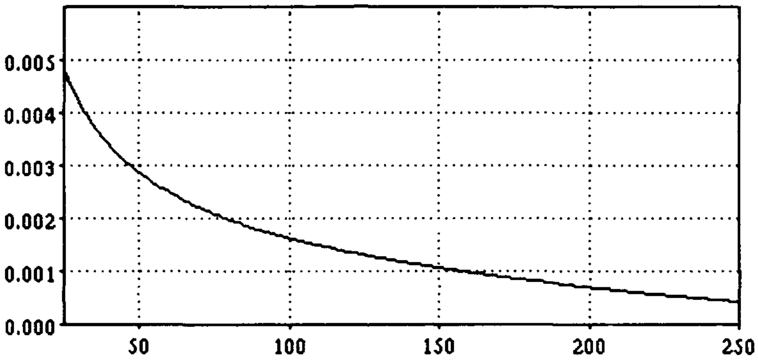

Figure 13:
Graph of selectivity $\mu$ for $\epsilon = 3$ as the basis vector length $d$ varies.

our attention to bases with the property that

$$\phi_0 \leq \phi \leq \pi - \phi_0 \qquad \text{or} \qquad \pi + \phi_0 \leq \phi \leq 2\pi - \phi_0.$$

By applying standard minimization methods, one finds that if the maximum distance between any two model points is $M$ and the minimum distance is $m$, then the maximum value for $\rho$ is given by

$$\rho_m \leq \frac{M}{m}\frac{1}{\sin\frac{\phi_0}{2}}. \tag{20}$$

To evaluate the selectivity, we also need to know $d$, the length of the basis vector, which can vary from 1 to $r$. Given a specific value for $d$, we can compute the selectivity. To get a sense of the variation of $\mu$ as $d$ changes, $\mu$ is plotted as a function of $d$ in Figure 13, for $\epsilon = 3$.

In general, $d$ will take on a variety of values, as the choice of basis points in the image is varied. To get an estimate for the expected degree of selectivity, we perform

| Case | Measured | Predicted | Approximation |
|---|---|---|---|
| $\epsilon = 1$ | .000116 | .000117 | .000118 |
| $\epsilon = 3$ | .001146 | .001052 | .001064 |
| $\epsilon = 5$ | .003142 | .002911 | .002955 |

Table 1:

Table comparing simulated and predicted selectivities. In all cases, the ratio of minimum to maximum separation of points was 10. The predicted column uses the full expression for $\overline{\mu}$ from equation 19, while the approximation column uses the approximation given by equation 23. The measured column reports actual observed selectivities obtained by generating sets of model and image features at random, and counting the number of matches, within error, for a pairing of an image and model basis.

the following analysis. We assume, for simplicity, that the origin of the image basis is at the center of the image. The second point used to establish the basis vector can in principle lie anywhere in the image, with equal probability. Hence the probability distribution for $d$ is roughly (ignoring corner effects in the image)

$$\frac{2d}{r^2}.$$

We could explicitly integrate equation 19 with respect to this distribution for $d$ to obtain an expected selectivity. This is messy, and instead we pursue two other options.

First, we can integrate this numerically for a set of examples, shown in Table 1 under the column marked *predicted*, which lists values for $\mu$ as a function of noise in the image (with an image dimension of $2r = 500$). The value of $\rho_m$ was set using $\phi_0 = \pi/16$, and a ratio of minimum to maximum model point separation of $M/m = 10$. It should be noted that varying $\phi_0$ over the range $\pi/8$ to $\pi/32$ produced results very similar to those reported in the table. As one would expect, the probability of a consistent match increases (selectivity decreases) with increasing error in the measurements. Thus we can see that for ranges of parameters that one would find in many recognition situations, a considerable fraction of the space of possible $\alpha$ and $\beta$ values are consistent with a given feature and basis.

To test the validity of our formal development, we ran a series of simulations on randomly chosen features to test the selectivity values $\mu$ predicted by equation (19). We generated sets of model and image features at random, chose bases for each at random, then checked empirically the probability that a model point, rewritten in the image basis, lay within the associated error disc of an image point. We chose to consider only cases in which the error disc fits entirely within the bounds of the image, since we know that our predictions are underestimates for the other cases. Table 1 summarizes the results, under the column marked *measured*.

Second, we can approximate the selectivity expression. By applying power series expansions for the different terms in equations 16, 17, 32 and 34, and keeping only first and second order terms, we arrive at

$$\mu \approx \frac{k\pi\epsilon^2}{r^2}\left[\frac{17}{12} + 2\log\frac{r}{d} + \frac{r^2-d^2}{rd}\right]. \tag{21}$$

Finding the expected value for equation 21 over the distribution for $d$, where $d$ can range from some minimum value $\ell$ to $r$, in turn yields the following approximation for the expected selectivity:

$$\overline{\mu} \approx \frac{2k\pi\epsilon^2}{r^2-\ell^2}\left[\frac{15}{8} - \frac{\ell^2}{r^2}\left(\frac{29}{24} + \log\frac{r}{\ell} + \frac{r}{\ell} - \frac{\ell}{3r}\right)\right]. \tag{22}$$

For the case of $\ell \ll r$, this reduces to

$$\overline{\mu} \approx 2k\pi\left(\frac{\epsilon}{r}\right)^2\left[\frac{15}{8} - \frac{\ell}{r} - \left(\frac{\ell}{r}\right)^2\log\frac{r}{\ell}\right] \tag{23}$$

and this predicts values in close agreement with those recorded in Table 1, as shown in the column marked *approximation.*

Note that the selectivity is clearly not linear in sensor error. For a fixed size image, increasing the error $\epsilon$ by some amount should decrease the selectivity (increase the probability) by at least a quadratic effect (perhaps more since there are higher order terms). This is reflected in Table 1, where increasing $\epsilon$ from 1 to 3 increases the predicted probability by roughly a factor of 9, and increasing from 1 to 5 increases the predicted probability by roughly a factor of 25. This expected value of the selectivity allows us to analyze the probability that a match will be reported at random by some recognition method that uses affine transformations. The selectivity, $\overline{\mu}$, in essence reflects the power of a given quadruple of features to distinguish a particular model. Now we consider the manner in which information from multiple quadruples is combined. This analysis differs slightly for different recognition methods. First we examine the geometric hashing method and then the alignment method.

## 4 The Geometric Hashing Method

We are now ready to investigate the probability that the geometric hashing method will randomly report a match of a model to an image, under an affine transformation from the model to the image [4, 16, 17, 18, 19, 22]. The geometric hashing method is based on the idea of representing an object by storing redundant transformation-invariant information about it in a hash table. At recognition time, similar invariants are computed from the sensory data, and are used to index into the hash table to find possible instances of the model. If enough of the sensor invariants score a hit when hashing against the model table, one has in principle found an instance of the model in the sensory data.

The formal description of the geometric hashing algorithm is for noise-free data. A number of variations of the basic geometric method have been presented, and have been illustrated using data from real images [16, 17, 18, 19, 22] with associated sensor noise. The experimental results reported in these papers, however, have been limited to relatively simple scenes. A modified version of the geometric hashing method has been reported in [4], where uncertainty in the image measurements is explicitly taken into account using a probabilistic model. This method addresses the issue of inexact sensor data, however it leaves open the question of formal characterizations of the expected performance of the method in the presence of noise and clutter.

Any hashing function should include an analysis of the conditions under which *collisions* will occur; when will different data items be mapped to the same key? In this section we provide such an analysis for the affine hashing method. This analysis is particularly crucial in the case of the affine hashing method, because the hash table is also implicitly used to allow for small amounts of uncertainty in the sensor data. That is, the method relies on the fact that 'similar' sensor values will be hashed to the same location. As we have seen briefly above, however, it is not possible to use any simple tessellation of the $\alpha$-$\beta$-space in order to correctly account for uncertainty. In particular, the range of $(\alpha, \beta)$ values consistent with a given point depends on the actual configuration of the model and image points. The configuration of the image points is not available at the time that the hash table is constructed, and thus a strictly correct table cannot be built. In practice, implementations of the method use approximations that simply tessellate the space uniformly and ignore the effects that this has both on false matches and false rejections.

## 4.1 Details of the Geometric Hashing Method

As with most model-based recognition methods, it is assumed that an object can be represented by a collection of features, or *interest points.* A 'match' of a model to a scene consists of a mapping of a subset of the model features to a subset of the image features, such that applying a geometric transformation of some particular type to all of the model features will make each of them coincident with their corresponding image feature. The geometric hashing approach has been used with various types of transformations, however here we restrict ourselves to the case of a two-dimensional affine transformation.

The geometric hashing method consists of two basic stages: (i) the construction of a model hash table and (ii) the matching of the models to an image. The hash table is used to store a redundant, transformation-invariant representation of each object. This representation makes use of the fact that a triple of model points defines an invariant coordinate frame, or *basis.* An affine-invariant model of an object is formed by expressing the locations of its feature points in terms of each such transformation-invariant coordinate frame (or basis), and using the resulting coordinates as indices for storing the corresponding basis in a hash table.

A key assumption underlying the method is what we will term the *affine hashing hypothesis*, which is that a point represented in terms of some basis will produce the same coordinate values under any valid transformation of that point and basis. This can be stated more formally as follows.

**Assumption 1** *Consider the four (ordered) 'model' points* $\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{m}_3$, *and* $\mathbf{m}_4$, *and the affine invariant coordinates* $(\alpha,\beta)$ *defined by* $\mathbf{m}_4 - \mathbf{m}_1 = \alpha(\mathbf{m}_2 - \mathbf{m}_1) + \beta(\mathbf{m}_3 - \mathbf{m}_1)$. *Let these four points undergo a transformation,* $T$, *and denote the resulting points by* $\mathbf{m}'_i = T\mathbf{m}_i, i = 1,\ldots,4$. *It is assumed to be the case that* $\mathbf{m}'_4 - \mathbf{m}'_1 = \alpha(\mathbf{m}'_2 - \mathbf{m}'_1) + \beta(\mathbf{m}'_3 - \mathbf{m}'_1)$.

When the transformation $T$ is an affine transformation, then this assumption is true [16, 17, 18, 19, 22]. However when $T$ is a transformation mapping a model to its image using a camera or other sensing device, there will generally be errors in the locations of the image points. In these cases the affine hashing assumption no longer holds. In the previous sections we have analyzed the extent to which uncertainty (or error) in the locations of image points impacts this assumption, and hence affects the geometric hashing method. In the following section we use this analysis to determine the probability that the affine hashing method will falsely report a match when none is present.

Before analyzing the performance of the method in the presence of sensor uncertainty, we describe the method assuming that there is no sensor error and no numerical roundoff error.

For each model, the following steps are used to enter it into the hash table:

1. Choose an ordered set of three model points $\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3$ as a basis, formed by an origin
$$\mathbf{o} = \mathbf{m}_1$$
and a pair of axes
$$\begin{aligned}\mathbf{u} &= \mathbf{m}_2 - \mathbf{m}_1\\ \mathbf{v} &= \mathbf{m}_3 - \mathbf{m}_1.\end{aligned}$$

2. For each additional model point $\mathbf{m}_i$, rewrite the coordinates of the vector $\mathbf{m}_i - \mathbf{o}$ in the affine basis defined by the axes $\mathbf{u}, \mathbf{v}$. In other words, find the coordinates $\alpha, \beta$ such that
$$\mathbf{m}_i - \mathbf{o} = \alpha\mathbf{u} + \beta\mathbf{v}.$$

3. Hash into a table using the indices $(\alpha,\beta)$, and store at that point in the table the basis triple $(\mathbf{o}, \mathbf{u}, \mathbf{v})$.

4. Repeat this process for all possible choices of model bases (that is for all ordered triples of model points). This results in a table indexed by affine-invariant coordinates. Any pair of $\alpha$ and $\beta$ values can be used to retrieve those model bases (if any) for which some model point $\mathbf{m}_i$ has the affine-invariant coordinates $(\alpha,\beta)$. In particular, if $(\alpha',\beta')$ are affine coordinates for

an image point, written in terms of some image basis, then $(\alpha',\beta') = (\alpha,\beta)$ if and only if there is a legal transformation of the four model points (the three basis points together with the point represented by the affine coordinates) that maps them onto the four associated image points.

At recognition time, the hash table is used to determine which models are present in the image. The idea is that if we select a triple of image points that corresponds to the model and compute the coordinates for other image features in terms of this basis, the hash table will contain corresponding entries because the model is stored in the table in terms of every possible basis (and the representation is invariant under any affine transformation). Thus, if we have selected an image basis that corresponds to the model, all the remaining image points that correspond to the model will produce $(\alpha,\beta)$ pairs that specify the same model basis in the hash table.

The exact processing at recognition time is as follows:

1. Choose a set of three sensor points $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3$ to form a basis, formed by an origin
$$\mathbf{O} = \mathbf{s}_1$$
and a pair of axes
$$\begin{aligned}\mathbf{U} &= \mathbf{s}_2 - \mathbf{s}_1\\ \mathbf{V} &= \mathbf{s}_3 - \mathbf{s}_1.\end{aligned}$$

2. For each additional sensor point $\mathbf{s}_i$, rewrite the coordinates of the vector $\mathbf{s}_i - \mathbf{O}$ in the affine basis defined by the axes $\mathbf{U}, \mathbf{V}$. In other words, find the coordinates $\alpha', \beta'$ such that
$$\mathbf{s}_i - \mathbf{O} = \alpha'\mathbf{U} + \beta'\mathbf{V}.$$

3. Index into the hash table using the indices $(\alpha',\beta')$, and retrieve the set of entries at that point in the table. Any bases stored at that location are possible candidate matches. Each time a given basis is retrieved from the table, a corresponding counter is incremented in a histogram. This step is repeated for all additional sensor points.

4. Once all the sensor points have been hashed, the histogram contains votes for those model bases that could correspond to the current sensor basis, $(\mathbf{O}, \mathbf{U}, \mathbf{V})$. If the peak in the histogram for a given model basis, $(\mathbf{o}, \mathbf{u}, \mathbf{v})$, is sufficiently high, then this basis is selected as a possible match. The entire model can then be transformed into the image coordinates and compared to verify that the hypothesized transformation is correct. The transformation from the model to the image coordinate frame can be computed from the corresponding model and image bases.

5. The entire operation is repeated for all possible bases (that is all triples of image points are considered until a match is found). On each iteration, the histogram counts are cleared.

Since the description of the algorithm is for perfect data, issues concerning the ellipse of uncertainty associated with a point are not addressed. In extending the method to deal with uncertainty, Step 3 must be modified, so that $(\alpha', \beta')$ are used to compute the ellipse of feasible values, and for any bucket in the hash table that intersects this ellipse, the stored entries are retrieve and used to increment the histogram. Presumably only one vote is cast for a model basis retrieved in this manner, even if it appears in more than one bucket overlapping the ellipse of uncertainty. Note, however, that because of these regions of uncertainty, a single model feature may be retrieved by more than one image feature, an issue to which we will return shortly.

## 5 The Sensitivity of Geometric Hashing in the Presence of Noise

Given that we can estimate ranges of values for the affine parameters $\alpha, \beta$, we can turn to the use of such ranges in examining the sensitivity of the geometric hashing method. The main question of concern is whether a random collection of sensor points can masquerade as a correct interpretation. That is, under what conditions is it likely that some random set of sensor points, rewritten with respect to some arbitrarily chosen sensor basis, will index an incorrect model basis enough times to give a histogram vote as large as the correct interpretation? We can investigate the probability of such false positive identifications with the following plan of action. (Recall that we analyze the case in which each point is correctly represented by an ellipse with a given uncertainty value, whereas the actual implementations of geometric hashing do nnt use this correct expression.)

1. We use the analysis from Section 4 to estimate the probability that a given quadruple of image points will match a given quadruple of model points, given bounded uncertainty of radius $\epsilon$ in the sensor data. We denote this by the selectivity $\overline{\mu}$ as given by equation (19), or its approximation in equation (23).

2. Each model basis is stored in the hash table according to the $m - 3$ remaining model features, and thus there are $m-3$ points that index to each model basis. We are interested in the probability that a randomly chosen image point and image basis will hash to a location in the $\alpha$-$\beta$-space that is consistent with a given model basis. This is just the probability that for at least one of the $m - 3$ model points, the image point lies within the error disc associated with the model point, rewritten in terms of the image basis. If we assume independently distributed features, then this is just

$$p = 1 - (1 - \overline{\mu})^{m-3}. \tag{24}$$

This follows from the fact that the probability that a particular model point is not consistent with a given pair of indices is $(1 - \overline{\mu})$ and by independence, the probability that all $m - 3$ points are not consistent with this pair of indices is $(1 - \overline{\mu})^{m-3}$. If this probability is reasonably large, then there is a high probability that we will see a large number of votes for a given basis at random. Note that in doing this, we are actually underestimating the probability of a vote. We should really evaluate the expected value of

$$\overline{1 - (1 - \mu)^{m-3}}$$

rather than just evaluating the expected value for $\overline{\mu}$ and using that directly in computing $p$. Doing so leads to a more complicated expression that gives values slightly larger than those obtained using the above expression. For simplicity, we use the expression in equation (24).

3. For each image basis, all $s - 3$ remaining image points (other than the 3 points used to establish the basis) are used to form an index to lookup corresponding bases into the table. If the probability of being consistent with a given basis is $p$, then $ps$ should be much smaller than $m$ if we are to avoid a false positive. More precisely, if the probability that a single hash lookup will cast a vote for a particular model basis is $p$, then the probability of exactly $k$ votes out of $s - 3$ is

$$q_k = \binom{s-3}{k} p^k (1 - p)^{s-3-k}. \tag{25}$$

Further, the probability of a false positive identification of size at least $k$ is

$$w_k = 1 - \sum_{i=0}^{k-1} q_i.$$

Note that this is the probability of a false positive for a particular sensor basis and a particular model basis.

4. Since the hash table is built by considering all possible model bases, there are $\binom{m}{3}$ different bases entered into the table. The probability of a false positive identification for a given sensor basis with respect to one model basis is $w_k$. Hence, the probability of a false positive for this given sensor basis with respect to any model basis is

$$e_k = 1 - (1 - w_k)^{\binom{m}{3}}. \tag{26}$$

## 5.1 Testing the model

To check the correctness of our model, we ran a series of experiments based on equation 25. In particular, we generated random sets of model and image features, with 25 model features and with $25, 50, 100$ and $200$ image features. We used our analysis to generate a predicted distribution for the probability of a false positive identification of size $k$. In each case, the uncertainty was set at $\epsilon = 3$, and the cutoff on angular stability was $\phi_0 = \frac{\pi}{16}$. These values, together with data about the minimum and maximum separation of model points were used to generate values for $\mu$, which were typically on the order of 0.0011.

For comparison, we also ran some simulations on these data sets, by selecting bases for both the model and the image at random, and determining the size of vote associated with that pairing of bases. In particular, for each additional model point, we computed the affine coordinates relative to the chosen basis, then used those coordinates to determine the nominal transformed position in the image. We also used those coordinates to determine the radius of the associated error disc. For each image point, we checked to see if at least one of the error discs about a transformed model point contained the image point. If so, we incremented the vote for this pairing of bases. This trial was repeated 1000 times. We excluded choices of model bases for which more than half the transformed points would lie outside the extent of the image. The results of these trials are shown in Figures (14) and (15).

One can see that the cases are in good agreement. In fact, our model tends to overestimate the probability of small false positives, and underestimate the probability of large false positives, so our results will tend to be conservative.

Next, we turn to the question of what $e_k$ looks like (recall that this is the probability that any sensor basis will have at least one matching model basis). As an illustration, we graph the probability of a false positive based on equation (26). In particular, we use a selectivity based on $\epsilon = 3, \phi_0 = \frac{\pi}{16}$, obtained from Table 1, and plot the value of $e_k$ for an object with 25 model features, for different values of $k$ and a given number of sensor features $s$. This is graphed in Figure (16). (A similar set of graphs, for $m = 38$ and $m = 50$, are also shown in Figure (16).) The process was repeated for different values for the number of sensor features $s$, generating the family of graphs in the figure. In Figure (17) we graph the same probability of a false positive based on equation (26), here using a selectivity corresponding to errors of $\epsilon = 5$.

In Figure (16) the correct interpretation cannot account for more than 22, 35 and 47 model features, respectively, because three of the $m$ features always match (and we used values of $m = 25, 38, 50$). Since in general the correct interpretation is likely to have fewer than this number of features due to occlusion, one can see that the probability of a false positive is "acceptable" only for cases with a moderate number of sensor features, and limited error. If the error bound is $\epsilon = 3$, then one can tolerate ratios of sensory data to model features as large as 10 : 1 while expecting with probability nearly 0 to have a false positive peak in the histogram as big as the model itself, for each choice of sensor basis. If we expect half of the model to be occluded then if the ratio of sensory data to model features is on the order of 5 : 1, we expect with probability nearly 1 to have a false positive peak in the histogram at least as big, but if the ratio is 3 : 1, the probability of a false peak is nearly 0.

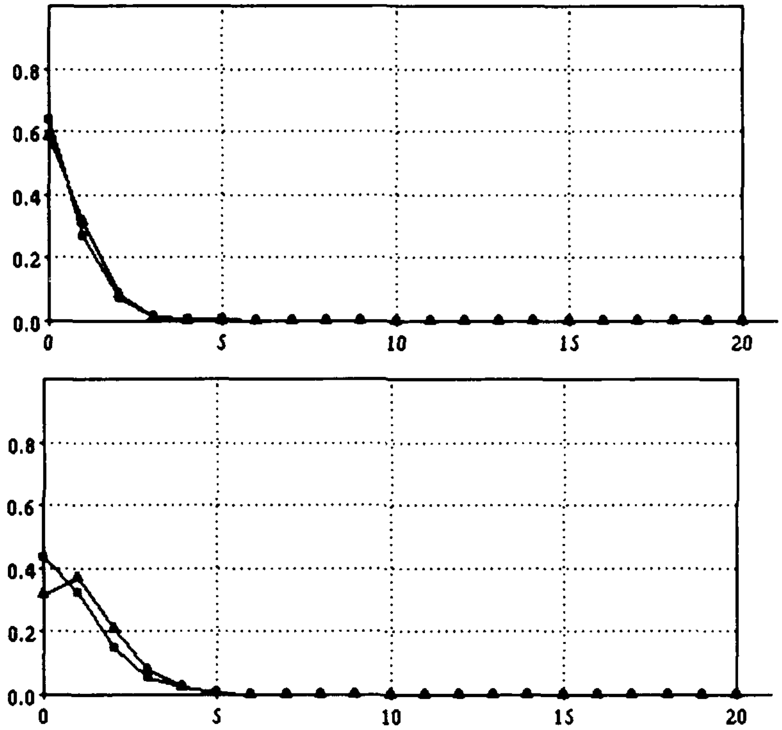

Figure 14:

Comparison of predicted and measured probabilities of false positives. Each graph compares the probability of a false peak of size $k$ observed at random. The cases are for $m = 25$ and $s = 25$ and 50, from top to bottom. In each case, $\epsilon = 3$, and $\phi_0 = \frac{\pi}{16}$. The graph drawn with triangles indicates the predicted probability, while the graph drawn with squares indicates the observed empirical probabilities.

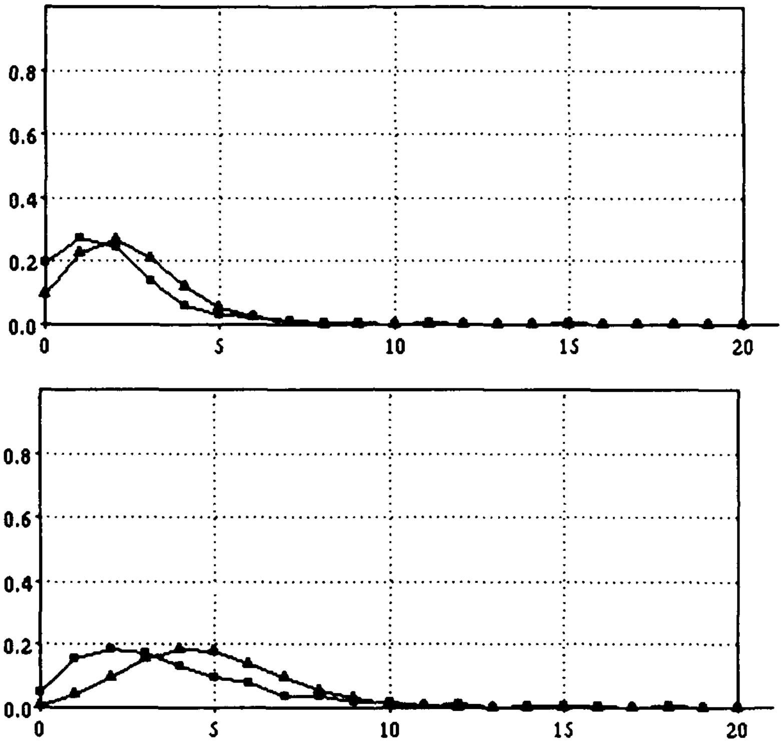

Figure 15:

Comparison of predicted and measured probabilities of false positives. Each graph compares the probability of a false peak of size $k$ observed at random. The cases are for $m = 25$ and $s = 100$ and 200, from top to bottom. In each case, $\epsilon = 3$, and $\phi_0 = \frac{\pi}{16}$. The graph drawn with triangles indicates the predicted probability, while the graph drawn with squares indicates the observed empirical probabilities.

When we consider errors of $\epsilon = 5$ (Figure 17), however, much less clutter can be tolerated. Now only ratios of sensor data to model features on the order of 4 : 1 can be tolerated while ensuring that the probability of a false peak as big as the model is nearly 0, and if the ratio is 6 : 1, this probability goes to nearly 1. If half of the model is occluded, then the corresponding ratios reduce to 1 : 1 and 2 : 1.

In Figure (18), we show the false positive rate as the error rate changes. Each figure plots the false positive rate, for model features $m = 25$, and for sensor features varying from $s = 25$ to $s = 200$ by increments of 25. The individual plots are for varying numbers of sensory features, and the process is repeated for changes in the bound on the sensor error, given a fixed threshold on angle of $\phi_0 = \pi/16$. One can see that if the error is very small, the method performs well, i.e. the probability of a false positive rapidly drops to zero even for small numbers of model features. As the sensor error increases, however, the probability of a false positive rapidly increases, as can be seen by comparing different families of plots in Figure (18). Note that the best possible correct solution would be for $k = 22$.

To compare our analysis with real data, we have performed the following test. Lamdan et al. [16] report data for the number of correct and incorrect votes for a model basis in the histogram, as a function of the size of the vote. This is done for an image with 28 features, and a model with 21 features. Using this data, we can estimate the probability of a false positive, for this image and model, as a function of the size of the vote. This is graphed in Figure (19), (the triangles). We can also use equation (26) to predict this probability. We do this for four different values for the selectivity factor, as indicated. One can see that while the graphs do not exactly match, due to the assumptions of the analysis, the predicted probability of a false positive is reasonably close to that observed in the real data case.

In part the results that we describe above are overly pessimistic, based on the error model of $\epsilon$-bounded sensory uncertainty. In essence this model assumes that the location of a feature within this error disc has a uniform probability for all positions within the disc. Perhaps a more realistic model would be to let the probability drop off with distance from the center of the disc, e.g. using a normal distribution. This is similar to the error model used in [4], who use a probabilistic formulation of positional uncertainty. We can model this effect with the following. Assume that while the overall bound on positional error is $\epsilon$, with a probability $\nu$, the deviation of the feature's position is $\epsilon'$. Then if $m$ is the number of model features, the expected size of the correct interpretation, given this error model, is

$$\nu o m \tag{27}$$

where $o$ is the fraction of the model expected to be occluded by other objects. For example, suppose $m = 25, o = .75$ and $\epsilon = 5$. Then the probability of a false positive of size 19 (the correct interpretation) is one, if $s = 25$ (see Figure 18c). On the other hand, if we consider, say, $\nu = .9$ for $\epsilon' = 1$, then we move from Figure 18c to Figure 18a. Now the expected size of the correct interpretation is 17, but we can tolerate sensor clutter as high as $s = 150$ and still have the probability of a false positive be vanishingly small.

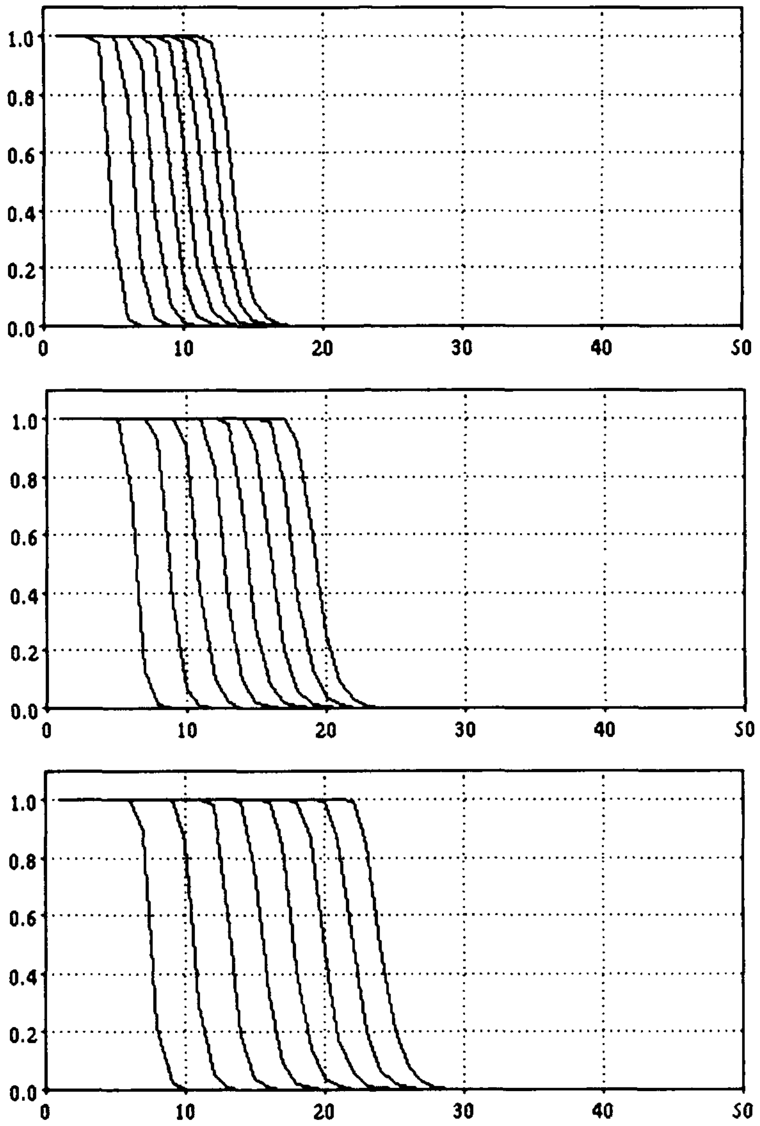

Figure 16:

Graph of probability of false positives. Vertical axis is probability of false positive of size $k$, horizontal axis is $k$. Each graph represents a different number of sensor features, starting with $s = 25$ for the left most graph, and increasing by increments of 25. In the top case, the model consisted of 25 features, in the middle case, 38 features, and in the bottom case, 50 features. Selectivity was based on $\epsilon = 3$ error.

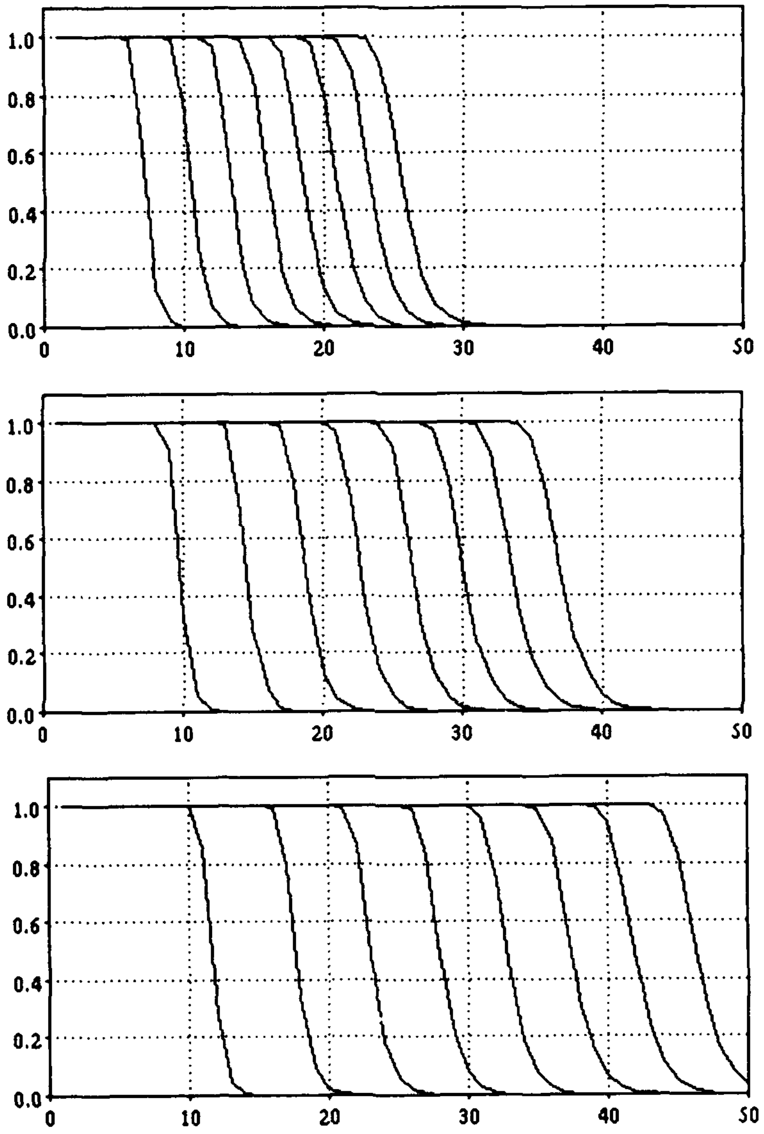

Figure 17:

Graph of probability of false positives. Vertical axis is probability of false positive of size $k$, horizontal axis is $k$. Each graph represents a different number of sensor features, starting with $s = 25$ for the left most graph, and increasing by increments of 25. In the top case, the model consisted of 25 features, in the middle case, 38 features, and in the bottom case, 50 features. Selectivity was based on $\epsilon = 5$ error.

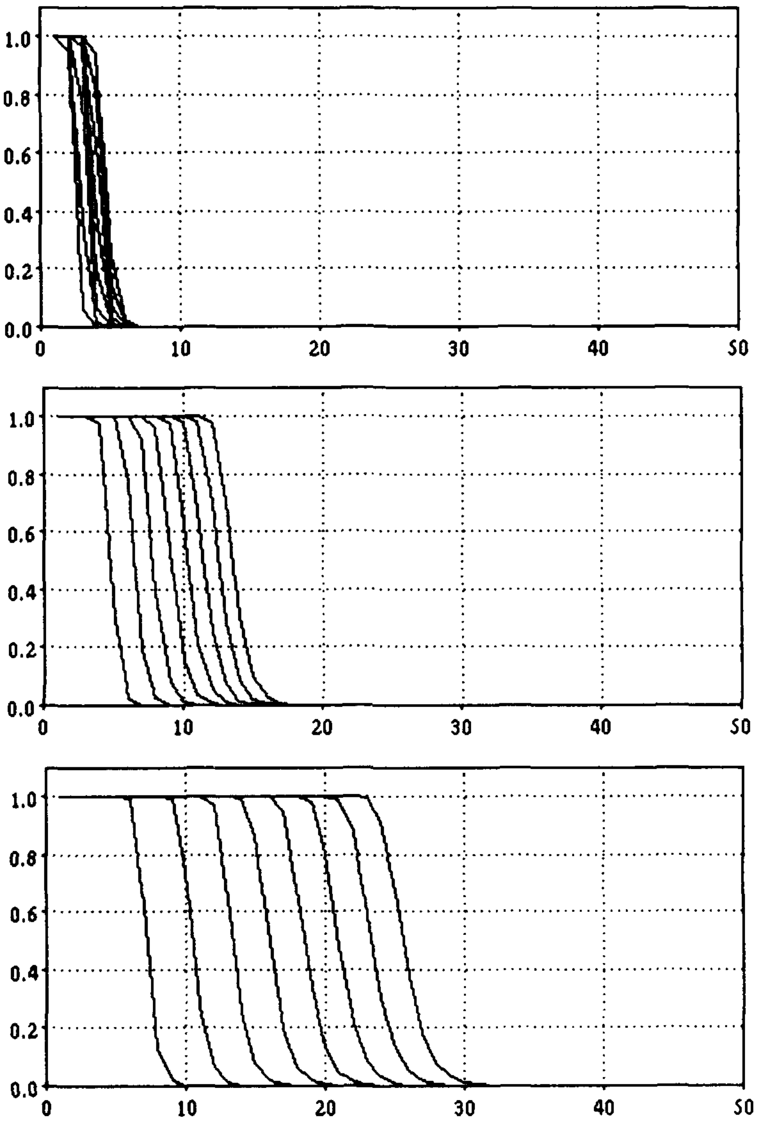

Figure 18:

Graph of probability of false positives. Vertical axis is probability of false positive of size $k$, horizontal axis is $k$. Each graph represents a different number of sensor features, starting with $s = 25$ for the left most graph, and increasing by increments of 25. In all cases, the model consisted of 25 features. In the top case, the sensor error was $\epsilon = 1$, in the middle case, $\epsilon = 3$, and in the bottom case, $\epsilon = 5$.

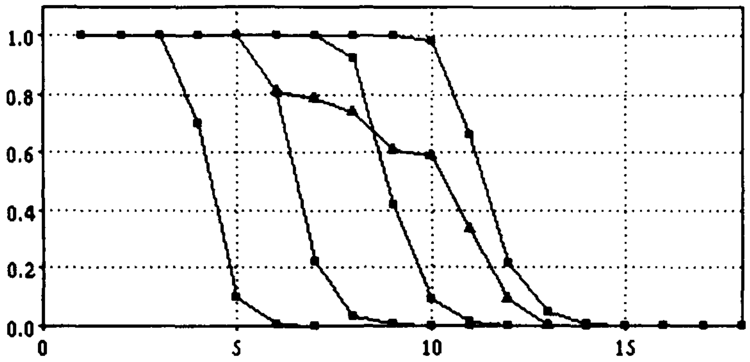

Figure 19:
Graph of probability of false positives - real data. Vertical axis is probability of false positive of size $k$, horizontal axis is $k$. The graph with the triangles is based on the data reported by Lamdan et al. for an image with $s = 28$ and a model with $m = 21$. The other three graphs (squares) are the predicted probability of a false positive, for selectivities based on $\epsilon = 3, 5, 7$, and 9 from left to right respectively.

In practice, the implementations of geometric hashing are using such an error model. By tesselating the hash table they are approximating an error region of some size (this is not exactly correct, since a square, or even circular, region in $\alpha$-$\beta$-space maps into an odd shaped region in image space). The size of this region is generally smaller than the actual bound on sensor error. This may cause the best interpretation to be smaller than one could achieve if one exactly modeled the error effects, but at the same time, it reduces the probability of seeing false positives.

# 6 The Sensitivity of Alignment in the Presence of Noise

A second object recognition method based on affine transformations is the alignment method [2, 12, 13]. The initial version of the affine-invariant alignment method was restricted to planar objects [12], whereas later versions operate on three-dimensional models (unlike affine hashing which uses two-dimensional models). The two-dimensional version of the alignment method bears some similarity to the geometric hashing approach, but differs in several fundamental aspects.

The basic alignment method is summarized as follows:

- Choose an ordered triple of image features and an ordered triple of model features, and hypothesize that these are in correspondence.
- Use this correspondence to compute an affine transformation mapping the model into the image.
- Apply this transformation to all of the remaining model features, thereby mapping them into the image.

- Search over an appropriate neighborhood about each projected model feature for a matching image feature, and count the total number of matched features.

This operation is in principle repeated for each ordered triple of model and image features, although it may be terminated after one or more matches are found, or after a certain number of triples are tried without finding a match.

The two-dimensional affine transformation, $A$, consisting of a linear transformation $\mathbf{L}$ and a translation $\mathbf{b}$, is computed from three pairs of model and image points $(\mathbf{a}_m, \mathbf{a}_i)$, $(\mathbf{b}_m, \mathbf{b}_i)$ and $(\mathbf{c}_m, \mathbf{c}_i)$, using the following procedure:

a) Translate the model so that the point $\mathbf{a}_m$ is at the origin.

b) Define the translation vector $\mathbf{b} = -\mathbf{a}_i$, and translate the image points so that th^ new $\mathbf{a}_i$ is at the origin, the new $\mathbf{b}_i$ is at $\mathbf{b}_i - \mathbf{a}_i$ and the new $\mathbf{c}_i$ is at $\mathbf{c}_i - \mathbf{a}_i$.

c) Solve for the linear transformation

$$\mathbf{L} = \begin{pmatrix} l_{11} & l_{12} \\ l_{21} & l_{22} \end{pmatrix}$$

given by the two pairs of equations in two unknowns

$$\mathbf{L}\mathbf{b}_m = \mathbf{b}_i,$$

and

$$\mathbf{L}\mathbf{c}_m = \mathbf{c}_i.$$

In a manner very similar to that used in the previous section, we can analyze the sensitivity of the alignment method. As before, the main question of concern is whether a random collection of sensor points can masquerade as a correct interpretation. In this case, we can investigate the probability of such false positive identifications with the following plan of action.

1. As before, the selectivity of a given quadruple of points is given by the expression for $\overline{\mu}$ in equation (19).

2. Since each model point is projected into the image, the probability that a given model point matches at least one image point is

$$p' = 1 - (1 - \overline{\mu})^{s-3}.$$

This follows from the fact that the probability that a particular model point is not consistent with a particular image point is $(1 - \overline{\mu})$ and by independence, the probability that all $s - 3$ points are not consistent with this model point is $(1 - \overline{\mu})^{s-3}$.

3. The process is repeated for each model point, so the probability of exactly $k$ of them having a match is

$$q'_k = \binom{m-3}{k}(p')^k(1-p')^{m-3-k}. \tag{28}$$

Further, the probability of a false positive identification of size at least $k$ is

$$w'_k = 1 - \sum_{i=0}^{k-1} q'_i.$$

Note that this is the probability of a false positive for a particular sensor basis and a particular model basis.

4. This process can be repeated for all choices of model bases, so the probability of a false positive identification for a given sensor basis with respect to any model basis is

$$e'_k = 1 - (1 - w'_k)^{\binom{m}{3}}. \tag{29}$$

## 6.1 Testing the model

To check the correctness of our model, we have run a series of experiments based on equation 28. In particular, we have used our analysis to generate a distribution for the probability of a false positive identification of size $k$, given $\epsilon = 3$ and $\phi_0 = \frac{\pi}{16}$, and using a model with 25 features and images with 25, 50, 100 and 200 features. For comparison, we also generated a set of model and image points of the same size, selected bases for each at random, and determined the size of vote associated with that pairing of bases. In particular, for each additional model point, we computed the affine coordinates relative to the chosen basis, then used those coordinates to determine the nominal position of an associated image point, together with a disc of uncertainty about that point. We simply checked to see if at least one image point lay within a model point's error disc. If so, we incremented the vote for this pairing of bases. This trial was repeated 1000 times. The results are shown in Figures (20) and (21).

One can see that the cases are in good agreement. In fact, our model tends to overestimate the probability of small false positives, and underestimate the probability of large false positives, so our results will tend to be conservative.

Next, we turn to the question of what $e'_k$ looks like. As an illustration, we graph in Figure (22) the probability of a false positive based on equation (29). In particular, we use a selectivity based on $\epsilon = 3$, obtained from Table 1, and plot the value of $e'_k$ for an object with 25 model features, for different values of $k$ and a given number of sensor features $s$. This is graphed in Figure (22). (A similar set of graphs, for $m = 38$ and $m = 50$, are also shown in Figure (22).) The process was repeated for different values for the number of sensor features $s$, generating the family of graphs in the figure.

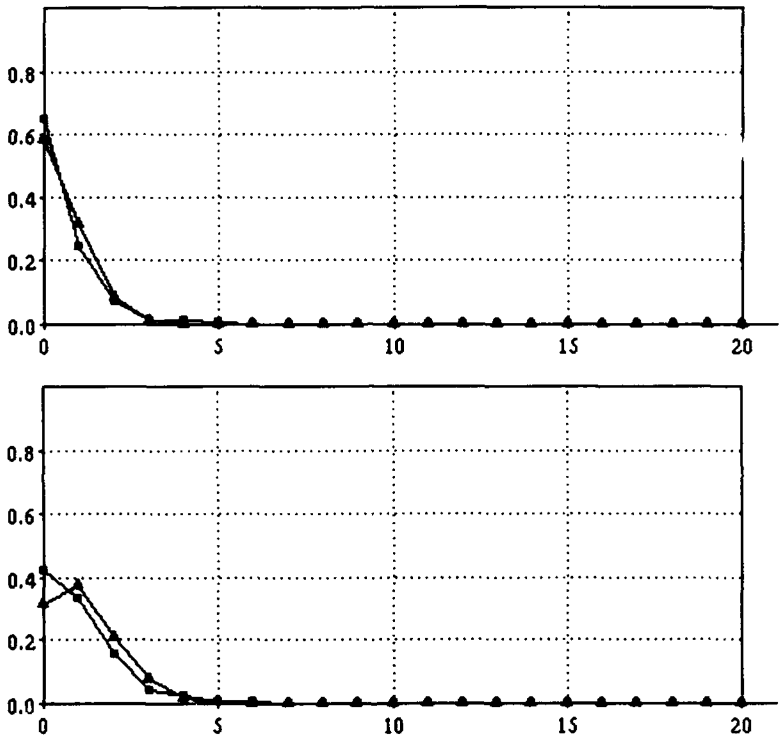

Figure 20:
Comparison of predicted and measured probabilities of false positives. Each graph compares the probability of a false peak of size $k$ observed at random. The cases are for $m = 25$ and $s = 25$ and 50, from top to bottom. In each case, $\epsilon = 3$.

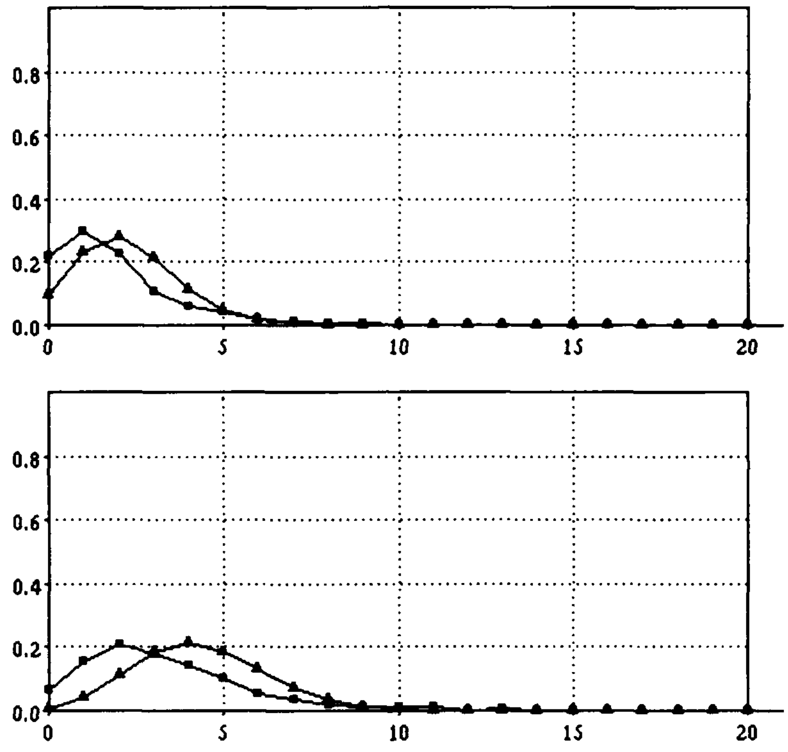

Figure 21:

Comparison of predicted and measured probabilities of false positives. Each graph compares the probability of a false peak of size $k$ observed at random. The cases are for $m = 25$ and $s = 100$ and 200, from top to bottom. In each case, $\epsilon = 3$.

In Figure (23) we graph the same probability of a false positive based on equation (29), here using a selectivity corresponding to errors of $\epsilon = 5$.

These results can be compared to the graphs for the geometric hashing method. Several observations are in order. First, the false positive curves are generally more favorable in the alignment case. That is, the probability of a false positive is considerably smaller for alignment than for the comparable case of hashing. This is mostly due to the fact that hashing is based on testing all image points to see if there is a matching model point, while alignment uses the model points to look for matching image points. Since there are generally many more image points than model points, alignment is likely to have a lower rate of false positives. If geometric hashing were to keep track of which model points have been matched to an image point, a comparable performance would be expected. Second, the false positive curves for the alignment approach an asymptotic limit of a step function, with cutoff at $m$, while geometric hashing, in the form considered here, tends to shift the curves linearly with increasing $s$. Overall, one concludes that alignment can tolerate considerably more clutter in the scene than hashing.

In Figure (24), we show the false positive rate, as the error rate changes. Each figure plots the false positive rate, for model features $m = 25$, and for sensor features varying from $s = 25$ to $s = 200$ by increments of 25. The individual plots are for varying numbers of sensory features, and the process is repeated for changes in the bound on the sensor error, given a fixed threshold on angle of $\phi_0 = \pi/16$. One can see that if the error is very small, the method performs well, i.e. the probability of a false positive rapidly drops to zero even for small numbers of model features. As the sensor error increases, however, the probability of a false positive rapidly increases, as can be seen by comparing different families of plots in Figure (24). Note that the best possible correct solution would be for $k = 22$.

# 7 Relation to Previous Work

The first analysis of the effects of sensor uncertainty on affine matching was done in [11]. At that time, we did not have the precise expression for error bounds given in Proposition 1, but we were able to produce approximations to the range of values for the affine coordinates $(\alpha, \beta)$. We used these in numerical simulations, showing empirically that the range of values associated with a pair of affine coordinates under this uncertainty model increased with increasing $\epsilon$, but also with the parameters associated with the basis vectors. Although the article only illustrated average selectivity values, obtained by averaging the selectivity ranges over all possible choices of points, our data supported the idea that selectivity also depended on the specific point, i.e. on the actual value of $(\alpha, \beta)$. The conclusion drawn in that work was that the verification stage of the recognition process would be critical for methods like geometric hashing, since a large number of possible pairings of model and image basis would be hypothesized. We also concluded that in order to keep the combinatorics manageable, geometric hashing, like generalized Hough transforms [9],

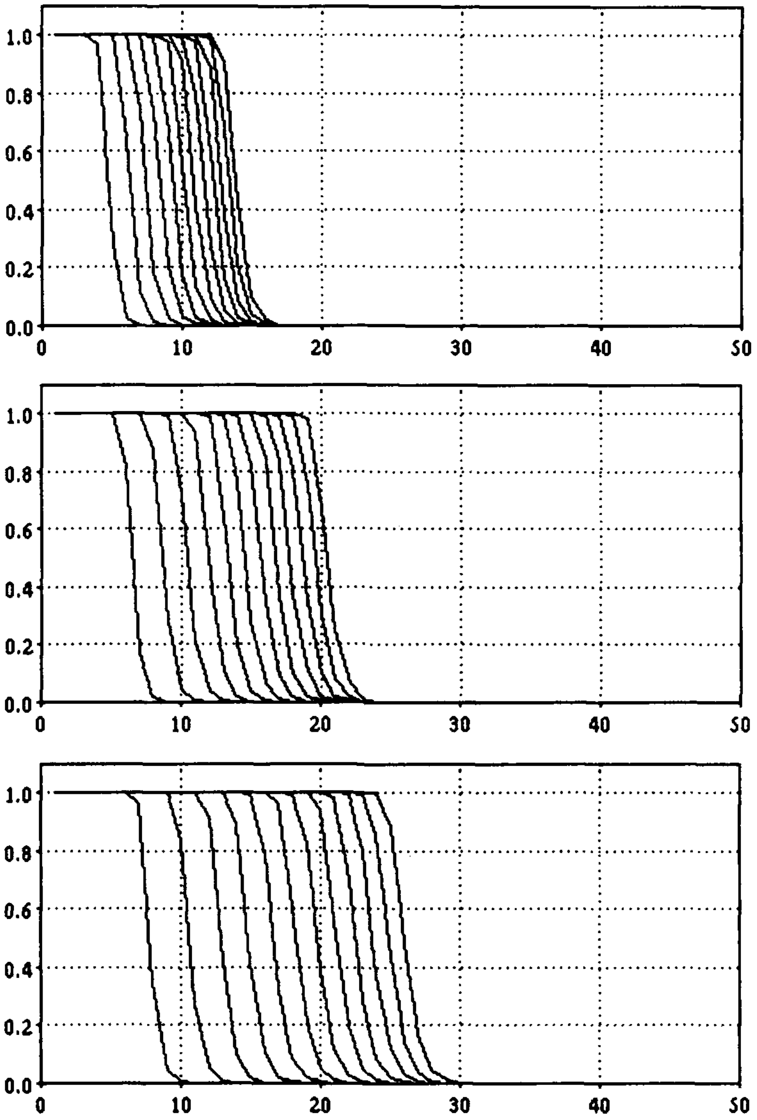

Figure 22:

Graph of probability of false positives. Vertical axis is probability of false positive of size $k$, horizontal axis is $k$. Each graph represents a different number of sensor features, starting with $s = 25$ for the left most graph, and increasing by increments of 25. In the top case, the model consisted of 25 features, in the middle case, 38 features, and in the bottom case, 50 features. Selectivity was based on $\epsilon = 3$ error.

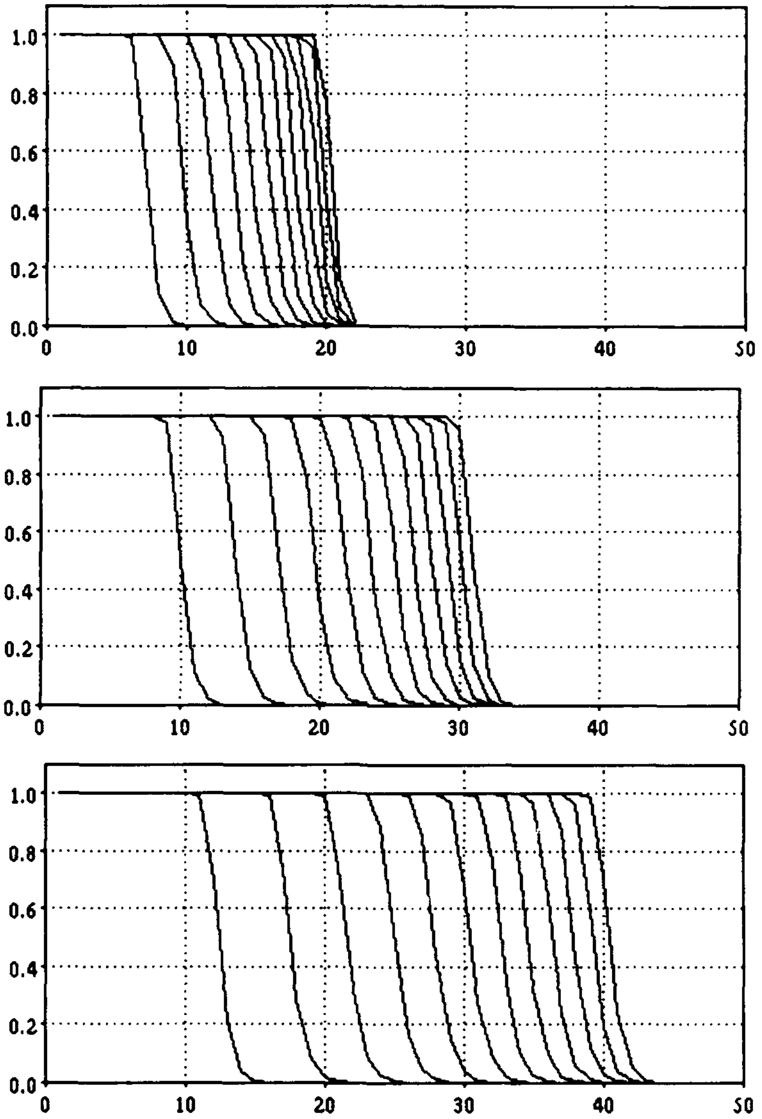

Figure 23:

**Graph of probability of false positives. Vertical axis is probability of false positive of size $k$, horizontal axis is $k$. Each graph represents a different number of sensor features, starting with $s = 25$ for the left most graph, and increasing by increments of 25. In the top case, the model consisted of 25 features, in the middle case, 38 features, and in the bottom case, 50 features. Selectivity was based on $\epsilon = 5$ error.**

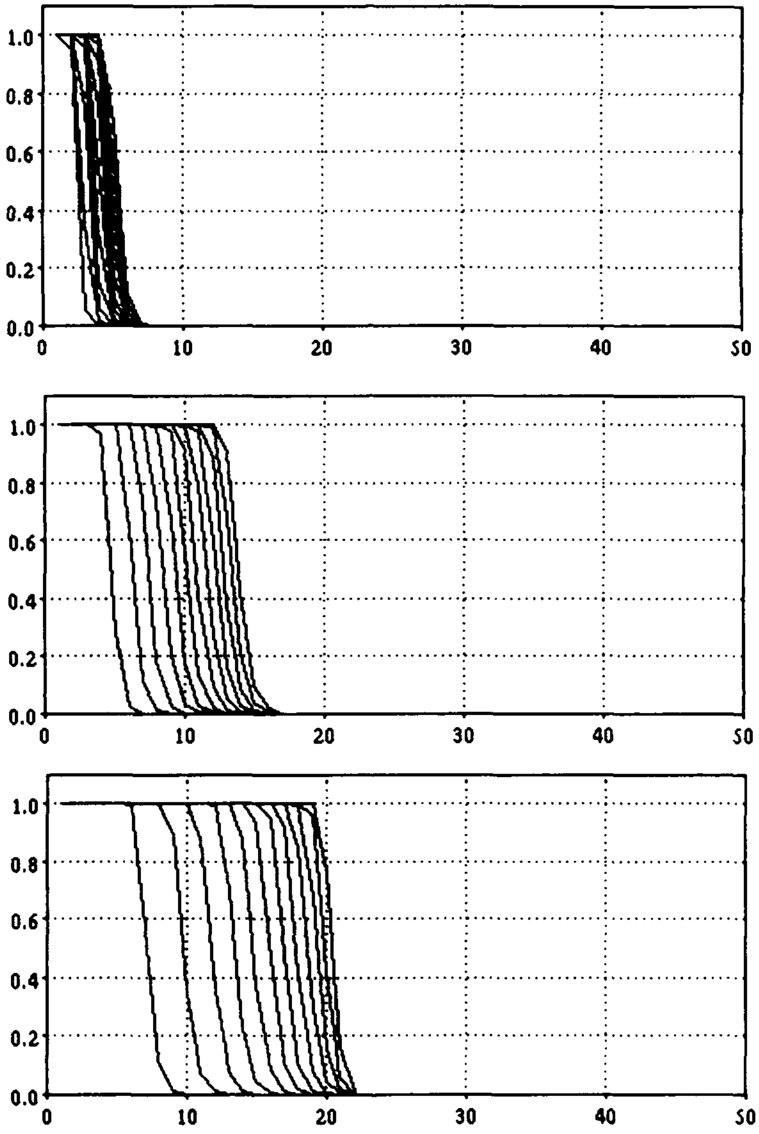

Figure 24:

Graph of probability of false positives. Vertical axis is probability of false positive of size $k$, horizontal axis is $k$. Each graph represents a different number of sensor features, starting with $s = 25$ for the left most graph, and increasing by increments of 25. In all cases, the model consisted of 25 features. In the top case, the sensor error was $\epsilon = 1$, in the middle case, $\epsilon = 3$, and in the bottom case, $\epsilon = 5$.

and tree search methods [7, 8], should be connected to a good grouping or selection method.

In a more recent analysis of the affine hashing method, Lamdan & Wolfson [20] (see also [22]) discuss the error properties of affine transformations under the same error model. Their approach is to consider the equation

$$\mathbf{A}\mathbf{x} = \mathbf{d}$$

where the columns of matrix $\mathbf{A}$ are defined by $\mathbf{m}_2 - \mathbf{m}_1$ and $\mathbf{m}_3 - \mathbf{m}_1$ and where $\mathbf{d} = \mathbf{p} - \mathbf{m}_1$, with $\mathbf{p}$ representing the point of interest. The vector $\mathbf{x}$ defines the affine coordinates of $\mathbf{p}$ and is obtained by inverting the matrix equation. To account for error, Lamdan & Wolfson consider

$$(\mathbf{A} + \delta\mathbf{A})(\mathbf{x} + \delta\mathbf{x}) = \mathbf{d} + \delta\mathbf{d}.$$

They claim that the values of the entries of $\delta\mathbf{A}$ and $\delta\mathbf{d}$ are bounded by $\epsilon$, and use this in their analysis. In fact, because these entries are defined as the difference of two uncertain vectors, the error bounds on the values should be $2\epsilon$ which suggests that the examples given in [20] are actually for cases in which the sensor error is half of that reported.

Based on this equation, they use results from numerical analysis to bound the magnitude of the uncertainty in the affine coordinates, using:

$$\frac{||\delta\mathbf{x}||}{||\mathbf{x}||} \leq \kappa(\mathbf{A}) \left[\frac{||\delta\mathbf{A}||}{||\mathbf{A}||} + \frac{||\delta\mathbf{d}||}{||\mathbf{d}||}\right] + O(\epsilon^2) \qquad (30)$$

where

$$\kappa(\mathbf{A}) = ||\mathbf{A}||\ ||\mathbf{A}^{-1}||$$

is the condition number of the matrix $\mathbf{A}$.

This result does not define the actual range of values, nor the shape of the region of uncertain values associated with an affine coordinate pair, but it does bound the magnitude of the uncertainty, in a manner roughly consistent with the results presented in [11]. Note that the magnitude of the uncertainty depends on the magnitude of the actual affine coordinates $||\mathbf{x}||$, as well as on the magnitude of the point $||\mathbf{d}||$ and properties of the affine basis $||\mathbf{A}||$. Using numerical simulations (modulo the incorrect values for $\epsilon$) Lamdan & Wolfson reach essentially the same conclusion as that of [11], namely that except in simple cases, geometric hashing without verification will produce far too many false hypotheses to be used as a pure recognition technique. They note, by using methods from [9], that in the case of rigid motions and simple scenes, verification may not be needed. They suggest that geometric hashing does form a useful preprocessing stage for subsequent verification, by reducing the number of cases that verification must consider.

# 8 Summary

The computation of an affine-invariant representation in terms of a coordinate frame $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$ has been used in the alignment [12, 13] and geometric hashing [16, 17, 18, 19, 22] model-based recognition methods. These recognition methods were both developed assuming no uncertainty in the sensory data, and then various heuristics were used to allow for error in the locations of sensed points. In this paper we have formally examined the effect of sensory uncertainty on these recognition methods. This analysis involves considering both the Euclidean plane used by the alignment method, and the space of affine-invariant $(\alpha,\beta)$ coordinates used by the geometric hashing method. Our analysis models each sensor point in terms a *disc* of possible locations, where the size of this disc is bounded by an uncertainty factor, $\epsilon$.

Under the bounded uncertainty error model, in the Euclidean space the set of possible values for a given point $\mathbf{x}$ and a basis $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$ forms a disc whose radius is bounded by $r = k\epsilon(1 + |\alpha| + |\beta|)$, where $1 \leq k \leq 2$. That is, assuming that each image point has a sensing uncertainty of magnitude $\epsilon$, the range of image locations that are consistent with $\mathbf{x}$ forms a circular region. In the $\alpha$-$\beta$-space, the set of possible values of the affine coordinates of a point $\mathbf{x}$ in terms of a basis $(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3)$ forms an ellipse (except in degenerate cases). The area, center and orientation of this ellipse are given by somewhat complicated expressions that depend on the actual configuration of the basis points.

The most important consequence of our analysis is the fact that the set of possible values in the $\alpha$-$\beta$-plane *cannot* be computed independent of the actual locations of the model or the image basis points. This means that the table constructed by the geometric hashing method can only approximate the correct values, because the locations of the image points are not known at the time that the table is constructed. We further find that the geometric hashing method works well when there is little noise in the measurements, and when the amount of spurious data in the scene is limited. When the noise levels are even moderate, however, we find that the method degrades considerably, and in particular, that the probability of a false positive recognition becomes significant. This probability also increases rapidly as a function of the number of sensory features. This suggests that the method will require that a substantial number of hypothesized matches be ruled out by some subsequent verification stage. In analyzing the alignment method, we find that the probability of a false match is substantially lower than for geometric hashing, largely because the alignment method explicitly keeps track of which model features have been matched to image features.

## A. Determining the Area of the Consistent Region

An approximation to the area in the third case, using the underestimate of Figure 10 with $p = \rho d$ leads to

$$\begin{aligned}
A_3 &= k \int_{c_1}^{c_2} 2\pi\epsilon^2(1+\rho)^2\rho^{-2}\,d\rho \\
&\quad + \int_{\rho=c_1}^{c_2} k(r-\rho d)\sqrt{4\epsilon^2(1+\rho)^2-(r-\rho d)^2}\rho^{-2}\,d\rho \\
&\quad + \int_{\rho=c_1}^{c_2} k2\epsilon(1+\rho)(r-\rho d)\rho^{-2}\,d\rho \\
&= 2\pi\epsilon^2 k\left[\rho + 2\log\rho - \frac{1}{\rho}\right]\Bigg|_{\rho=c_1}^{c_2} \\
&\quad + k\left[-\left(\frac{r}{\rho}+d\right)\sqrt{a+b\rho+c\rho^2}\right. \\
&\qquad + \left(\frac{br}{2}-ad\right)\frac{1}{\sqrt{-a}}\arcsin\frac{2a+b\rho}{\rho\sqrt{-\Delta}} \\
&\qquad \left. - \left(cr-\frac{bd}{2}\right)\frac{1}{\sqrt{-c}}\arcsin\frac{2c\rho+b}{\sqrt{-\Delta}}\right]\Bigg|_{\rho=c_1}^{c_2} \\
&\quad + 2\epsilon k\left[\frac{-r}{\rho}+(r-d)\log\rho - d\rho\right]\Bigg|_{\rho=c_1}^{c_2} \qquad (31)
\end{aligned}$$

where

$$\begin{aligned}
a &= 4\epsilon^2 - r^2 \\
b &= 8\epsilon^2 + 2dr \\
c &= 4\epsilon^2 - d^2 \\
\Delta &= 4ac - b^2.
\end{aligned}$$

Here $c_2 = \min\left\{\rho_m, \frac{r}{d}\right\}$, so that we integrate out until either we hit the maximum value for $\rho$, or the center of the disc reaches the edge of the image. Substitution, under the condition that $\rho_m > c_2$ yields

$$\begin{aligned}
A_3 = 2k &\left[\frac{dr^2+2\epsilon^2(r-d)}{\sqrt{r^2-4\epsilon^2}}\left(\arcsin\frac{2\epsilon}{r}+\frac{\pi}{2}\right)\right. \\
&+ \frac{d^2r+2\epsilon^2(d-r)}{\sqrt{d^2-4\epsilon^2}}\left(\arcsin\frac{2\epsilon}{d}-\frac{\pi}{2}\right) \\
&+ \epsilon(r-d)\left(\log\frac{r}{d}-\log\frac{r-2\epsilon}{d+2\epsilon}\right) \\
&+ \frac{2\epsilon(d+r)\left(3\epsilon(d-r)-rd+8\epsilon^2\right)}{(r-2\epsilon)(d+2\epsilon)} \\
&\left. + 2\pi\epsilon^2\left[\epsilon(r+d)\frac{r^2+d^2+2\epsilon(d-r)}{dr(d+2\epsilon)(r-2\epsilon)}+\log\frac{r}{d}-\log\frac{r-2\epsilon}{d+2\epsilon}\right]\right]. \qquad (32)
\end{aligned}$$

An approximation to the area in the fourth case, using the approximation shown in Figure 11 yields:

$$\begin{aligned} A_4 &= \int_{\rho=c_2}^{c_3} k\left(2\epsilon(1+\rho)-\rho d+r\right)\sqrt{4\epsilon^2(1+\rho)^2-(r-\rho d)^2}\rho^{-2}\,d\rho \\ &= k\left[\left(2\epsilon-d-\frac{2\epsilon+r}{\rho}\right)\sqrt{a+b\rho+c\rho^2}\right. \\ &\qquad +\left(\frac{b(2\epsilon+r)}{2}+a(2\epsilon-d)\right)\frac{1}{\sqrt{-a}}\arcsin\frac{2a+b\rho}{\rho\sqrt{-\Delta}} \\ &\qquad \left.-\left(c(2\epsilon+r)+\frac{b}{2}(2\epsilon-d)\right)\frac{1}{\sqrt{-c}}\arcsin\frac{2c\rho+b}{\sqrt{-\Delta}}\right]\Bigg|_{\rho=c_2}^{c_3} \end{aligned} \tag{33}$$

where $c_3 = \min\{\rho_m, \frac{r+2\epsilon}{d-2\epsilon}\}$.

Substitution, under the condition that $\rho_m > c_3$ yields

$$\begin{aligned} A_4 &= 2k\left[2\epsilon(d+r)\frac{dr+\epsilon(d-r)}{dr}\right. \\ &\qquad +\frac{8\epsilon^3+dr^2+\epsilon(r-2\epsilon)(d-r)}{\sqrt{r^2-4\epsilon^2}}\left(\frac{\pi}{2}-\arcsin\frac{2\epsilon}{r}\right) \\ &\qquad \left.+\frac{8\epsilon^3-d^2r-\epsilon(d+2\epsilon)(d-r)}{\sqrt{d^2-4\epsilon^2}}\left(\frac{\pi}{2}+\arcsin\frac{2\epsilon}{d}\right)\right]. \end{aligned} \tag{34}$$

# References

[1] Ballard, D.H., 1981, "Generalizing the Hough Transform to Detect Arbitrary Patterns," *Pattern Recognition* **13**(2): 111–122.

[2] Basri, R. & S. Ullman, 1988, "The Alignment of Objects with Smooth Surfaces," *Second Int. Conf. Comp. Vision*, 482–488.

[3] Besl, P.J. & R.C. Jain, 1985, "Three-dimensional Object Recognition," *ACM Computing Surveys*, Vol. 17, no. 1, pp. 75–154.

[4] Costa, M., R.M. Haralick & L.G. Shapiro, 1990, "Optimal Affine-Invariant Point Matching," *Proc. 6th Israel Conf. on AI*, pp. 35–61.

[5] Cyganski, D. & J.A. Orr, 1985, "Applications of Tensor Theory to Object Recognition and Orientation Determination", *IEEE Trans. on Pat. Anal. and Mach. Intel.*, Vol. PAMI-7, no. 6, pp. 662–673.

[6] Efimov, N.V., 1980, *Higher Geometry*, translated by P.C. Sinha. Mir Publishers, Moscow.

[7] Grimson, W.E.L., 1990, *Object Recognition by Computer: The role of geometric constraints*, MIT Press, Cambridge.

[8] Grimson, W.E.L., 1990, "The Combinatorics of Heuristic Search Termination for Object Recognition in Cluttered Environments," *First Europ. Conf. on Comp. Vis.*, pp. 552–556.

[9] Grimson, W.E.L. & D.P. Huttenlocher, 1990, "On the Sensitivity of the Hough Transform for Object Recognition," *IEEE Trans. PAMI* **12**(3), pp. 255–274.

[10] Grimson, W.E.L. & D.P. Huttenlocher, 1991, "On the Verification of Hypothesized Matches in Model-Based Recognition", to appear in *IEEE Trans. on Pattern Analysis and Machine Intelligence.*

[11] Grimson, W.E.L. & D.P. Huttenlocher, 1990, "On the Sensitivity of Geometric Hashing", *Proceedings of the Third International Conference on Computer Vision*, pp. 334–338.

[12] Huttenlocher, D.P. and S. Ullman, 1987, "Object Recognition Using Alignment", *Proceedings of the First International Conference on Computer Vision*, pp. 102-111.

[13] Huttenlocher, D.P. & S. Ullman, 1990, "Recognizing Solid Objects by Alignment with an Image," *Inter. Journ. Comp. Vision* **5**(2):195–212.

[14] Klein, F., 1939, *Elementary Mathematics from an Advanced Standpoint: Geometry*, MacMillan, New York.

[15] Korn, G.A. & T.M. Korn, 1968, *Mathematical Handbook for Scientists and Engineers*, McGraw-Hill, New York.

[16] Lamdan, Y., J.T. Schwartz & H.J. Wolfson, 1988a, "On Recognition of 3-D Objects from 2-D Images," *IEEE Int. Conf. on Rob. Aut.* pp. 1407–1413.

[17] Lamdan, Y., J.T. Schwartz & H.J. Wolfson, 1988b, "Object Recognition by Affine Invariant Matching," *IEEE Conf. on Comp. Vis. and Patt. Recog.* pp. 335–344.

[18] Lamdan, Y., J.T. Schwartz & H.J. Wolfson, 1990, "Affine Invariant Model-Based Object Recognition," *IEEE Trans. Robotics and Automation*, vol. 6, pp. 578–589.

[19] Lamdan, Y. & H.J. Wolfson, 1988, "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," *Second Int. Conf. on Comp. Vis.* pp. 238–249.

[20] Lamdan, Y. & H.J. Wolfson, 1991, "On the Error Analysis of 'Geometric Hashing'," *IEEE Conf. on Comp. Vis. and Patt. Recog.* pp. 22–27.

[21] Thompson, D. & J.L. Mundy, 1987, "Three-Dimensional Model Matching From an Unconstrained Viewpoint", *Proc. IEEE Conf. Robotics and Automation* p. 280, IEEE Computer Society Press.

[22] Wolfson, H.J., 1990, "Model Based Object Recognition by Geometric Hashing," *First Europ. Conf. on Comp. Vis.* pp. 526–536.